In presenting the dissertation as a partial fulfillment of the requirements for an advanced degree from the Georgia Institute of Technology, I agree that the Library of the Institute shall make it available for inspection and circulation in accordance with its regulations governing materials of this type. I agree that permission to copy from, or to publish from, this dissertation may be granted by the professor under whose direction it was written, or, in his absence, by the Dean of the Graduate Division when such copying or publication is solely for scholarly purposes and does not involve potential financial gain. It is understood that any copying from, or publication of, this dissertation which involves potential financial gain will not be allowed without written permission.

3/17/65
b

DEVELOPMENT OF AUTOMATED OPTIMUM
STRUCTURAL DESIGN SYSTEMS

A THESIS

Presented to

The Faculty of the Graduate Division

by

Marion Willson Corey

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

in the School of Civil Engineering

Georgia Institute of Technology

March, 1968

DEVELOPMENT OF AUTOMATED OPTIMUM

STRUCTURAL DESIGN SYSTEMS

Approved:

____

____

____

Date approved by Chairman: March 6, 1968

ACKNOWLEDGEMENTS

Grateful acknowledgement is expressed to Dr. F. W. Schutz, Jr. for his advice and counsel in this investigation. Sincere appreciation is extended to Mr. Gerald A. Matthews, Director of the Mississippi State University Computing Center, for the liberal use of the computing facilities used in this research.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF ILLUSTRATIONS

SUMMARY

Structural design presents a formidable calculation task to the engineer. In the late 1950's civil engineers began to use computers for the solution of computational problems. Later analysis programs were developed by integrating the computational programs. More recently the trend in engineering computer utilization has been to provide the engineer with a means of communicating with general purpose analysis programs in a familiar language. Typical of these programs are COGO, STRESS, and ICES. The use of these programs in the design process has been referred to as "computer-aided" design. In this environment the engineer would make design decisions, relay these decisions to the computer, and allow the computer to analyze his design.

Structural design is an iterative process. At the end of each iteration either the trial design must be modified if the specifications have not been met, or it may be modified in an attempt to obtain a more economical design. A design which meets the specifications and which cannot be improved economically is the optimum design. This process may be automated through the use of recently developed optimization procedures if an objective function and the limits of the design parameters can be stated.

This thesis presents a general procedure for the development of automated optimum structural design systems. Also presented is an automated optimum design system for single span, variable section, steel frames.

The procedure for the development of an automated optimum structural design system is based on the classification of the variables and parameters of structural design problems. Computer routines must be written for each classification. The classifications used are problem variables, problem constraints, design parameters, and design constraints.

Problem variables specify the requirements of the problem while problem constraints are the limitations within which the problem requirements must be met. Problem variables together with the problem constraints include all those items necessary for a complete definition of the structural design problem. They are generally determined by the owner, the architect, building codes, specifications, or vendors.

Design parameters and design constraints are controlled by the engineer. Design parameters are those items which when assigned values reduce the design problem to one of linear selection for the remaining items, which are then defined as design constraints. The modes of variation of the design parameters and constraints are referred to as constant, array, incremental, or continuous.

The goal of the structural design process is stated as follows: From the problem constraints select that set of design parameters with its associated design constraints which satisfies the problem requirements and optimizes the objective function.

Initialization and input computer routines are required for the problem variables and constraints. The initialization routine supplies common values as default options. The input routine provides for user communication with the system. Use of keywords and default options will eliminate the need for rigid input requirements.

Optimizing routines are used for the design parameters. The functions of these routines are to set the initial values of the design parameters, change the values of the design parameters, and determine when the optimum values of the design parameters have been obtained.

The function of the routines handling the design constraints is to determine if the specifications have been met and if so would a smaller value also satisfy the specifications. If the specifications are not met then a larger value is used if available. If no larger value is available then the set of design parameters is marked as a non-feasible solution. These routines together with an analysis routine may be organized according to the outline in this work into successful automated optimal design systems.

Optimum searching methods allow the optimization of functions whose analytical form is unknown. Since the objective function of a structural design problem usually cannot be stated directly in terms of the design parameters, these methods will be used. Optimum searching subroutines applicable to structural design problems are presented. These subroutines use two search plans, namely, orthogonal and diagonal searching. Orthogonal searching is the orthogonal application of one dimensional search techniques to multi-dimensional problems. The one dimensional techniques used are the golden section search for continuous variation and lattice search for incremental or array variation. Diagonal searching is used to check the points on the non-orthogonal vectors adjacent to the optimum indicated by orthogonal searching.

Variable classifications for the single span, variable section steel frame design problem are listed along with the names and functions of the subroutines required for each of the classifications. The complete FORTRAN source listing for this system along with an example problem listing is included. Thirty-one control cards are provided by this system for problem specification. Only three cards are absolutely necessary to the system. Where cards are omitted the system will supply commonly used values for the variables. These control cards give the user complete control over problem specification, type of output and mode of parameter variation. The general form of each of the control cards along with their options, units, and default values is presented.

Use of optimum searching techniques implies that the objective function is unimodal. The system presented in this work has been used to examine the variation of the objective function. It was found that discontinuities and local minima often occur in the objective function due to discrete changes in the design constraints as the design parameters are varied. The search techniques used allow the objective function to be examined only at discrete intervals. When examined in this fashion the function generally appears to be unimodal.

It is concluded that:

(a) Variables and parameters of structural design problems can be classified according to who or what controls their values and computer routines written for each classification.

(b) These routines together with an analysis routine can be organized according to the outline presented to create successful automated optimum structural design systems.

(c) Discontinuities and local minima may occur in the objective function due to discrete changes in the design constraints as the design parameters are varied.

(d) Objective functions of structural design problems generally demonstrate a sufficiently convex trend to allow successful use of optimum searching techniques.

CHAPTER I

INTRODUCTION

Historical Review

Structural design presents a formidable calculation task to the engineer. The process of performing these calculations has advanced from longhand and logarithms to the slide rule, desk calculator, and finally to the electronic computer (6). In the late 1950's civil engineers began to use computers for solution of the repetitive calculations required by classical methods of design. Computer programs available at the time were able to execute only particular computational tasks such as finding the latitudes and departures of a traverse, performing moment distribution, or computing stiffness coefficients of variable section beams. Later these independent calculation programs were integrated into analysis programs (8). Typical programs of this era covered such topics as traverse closure, rigid frame analysis, truss analysis, and water network distribution analysis.

More recently the trend in engineering computer utilization has been to provide the engineer with a means of communicating with general purpose analysis programs in a familiar language (9). Typical of these programs in the civil engineering field are COGO, STRESS, and ICES (3,7,11).

The use of general analysis programs such as STRESS or COGO in the design process requires the transfer of a large amount of information between the engineer and the computer, and vice versa. The role of the engineer in this environment would be to make design decisions, relay these decisions to the computer, and allow the computer to analyze his design. This process has been referred to as computer-aided design (12). Computer-aided design implies use of a general analysis program such as STRESS or STRUDL by a design engineer. The engineer's role is, first, to specify an initial design and submit it to the computer program for analysis. Second, acting on output from the analysis program, the engineer decides on a course of action to improve the design, and submits appropriate design changes to the program for analysis of the modified design. The responsibility for intelligent design judgment is solely that of the engineer, and the design is repeatedly refined until the process is terminated by the engineer.

Structural design is an iterative process requiring that one of two types of decisions be made at the end of each iteration. The trial design must be modified if the specifications have not been met, or it may be modified in an attempt to obtain a more economical design. A design which meets the required specifications and which cannot be improved economically is then the optimum design. Automation of this process requires that some type of optimizing routine replace the engineer as the decision maker. If some objective function can be written under which the merit of a particular trial design might be evaluated, and if the constraints and the limits of variation of the design parameters can be stated, then some type of optimization procedure might indeed be used.

Until recently the only known methods of handling optimization problems were the classical differential and variational calculus. With the rise of "operations research" other methods of optimization have been developed (14). These methods include such techniques as linear programming, non-linear programming, dynamic programming, and optimum seeking procedures. Structural designers are now beginning to use these methods. Grid searches and dynamic programming techniques have been used in designing plate girders (10). Non-linear programming methods have been used to aid in the design of prestressed concrete beams (4). Linear programming techniques have been used in the selection of least weight members for plastically designed frames (2,13).

The objective function of most structural design problems cannot be stated directly in terms of the design parameters. The design engineer then must experiment in order to find its optimum value and the associated values of the design parameters. Optimal seeking procedures make possible the optimization of a function whose analytical form is unknown. Thus these procedures may contribute to the automation of structural design problems.

In January 1966 the Structural Division Research Committee of the Americal Society of Civil Engineers stated:

> Present efforts to increase the usefulness of electronic computers to structural engineers by the generation of simplified computer languages (such as FORTRAN) and programs (such as STRESS and STRUDL) must be continued. However, probably the single most significant advancement in structural engineering would be the development of complete computer design processes. It is not unreasonable to imagine that, in time, an engineer need only specify loads, general geometric form, and controlling geometric coordinates for a structure, and then rely on the computer to carry out the design, including the preparation of drawings, optimizing the design within the framework of any

> selected set of specifications or other restrictions that need be imposed. (1)

The committee estimates that a reasonable research program in this area would cost $2.5 million per year for at least the next decade. Among the benefits which would accrue from such a capability would be a reduction in the time required to prepare designs and an increased ability to consider alternate designs. It is suggested that a system such as proposed by the Structural Division Research Committee might finally be an integration of many sub-systems, each for a particular geometric form.

Objective

It is the primary objective of the present work to present a general method for the development of automated optimum structural design systems. This method will be used to develop an automated optimal design system for a structure of a particular geometric form.

A flow chart of the structural design process is shown in Figure 1. The most tedious and time consuming portion of this process is involved in the analysis, the quantitative evaluation, and the changing of parameters to satisfy the design requirements or to optimize the objective function. Computer programs now exist which will handle the analysis portion of this process.

This work will illustrate how systems can be developed to handle the quantitative evaluation and the changing of parameters to arrive automatically at an optimum design. The parameters common to structural design problems will be examined so that they may be classed and assigned to the various sections of the design process for responsibility. The modes of possible variation of these parameters will be examined so that optimizing routines suitable to these modes may be suggested. Finally, since systems such as these are developed for the utility of the engineer, an attempt should be made to make their use as effortless and as unrestrictive as possible. Several methods which have been developed by recent studies in man-machine communication will be suggested as ways to accomplish this.

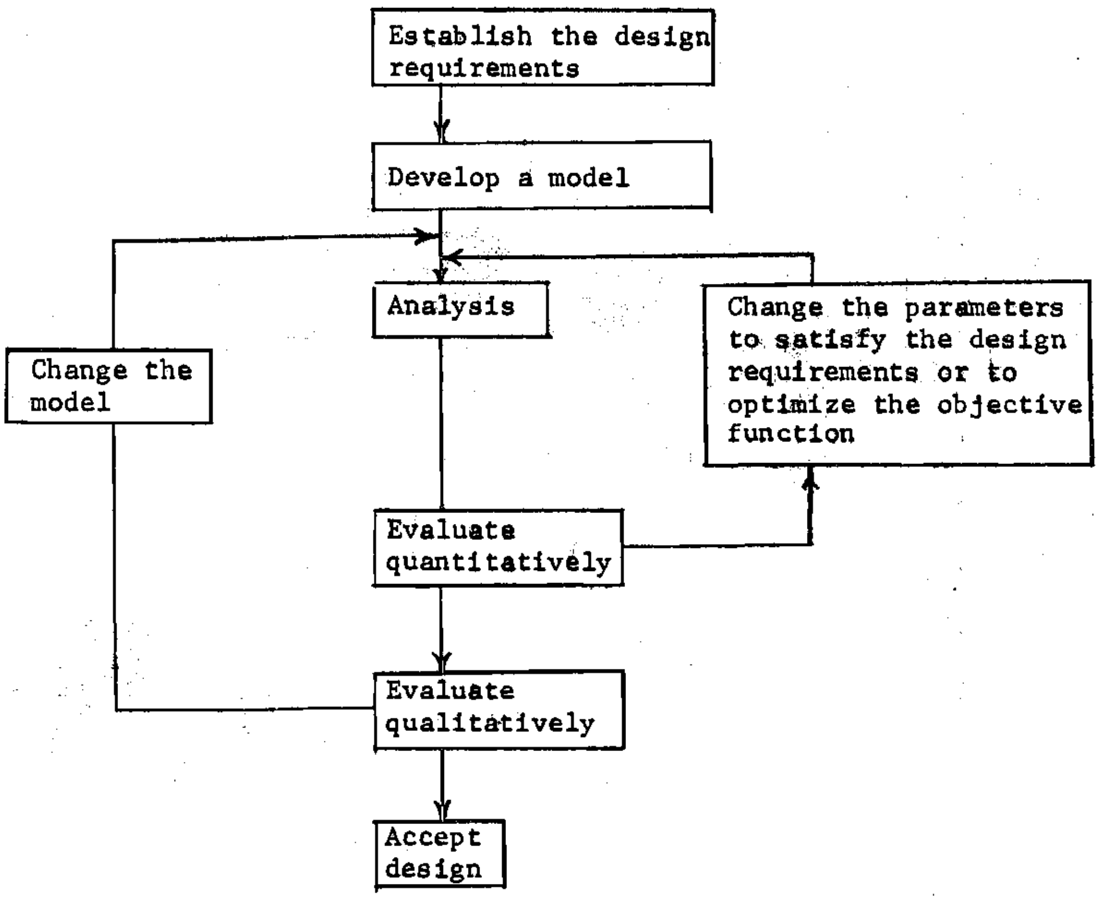


Figure 1. Simplified Flow Chart of the Structural Design Process

CHAPTER II

PROCEDURE

Organization of the Structural Design Process

In order to organize a structural design procedure it is necessary to classify the problem variables and parameters. The following classifications will be used here.

1. Problem Variables
2. Problem Constraints
3. Design Parameters
4. Design Constraints

Problem variables specify the requirements of the problem. They include such items as heights of stories, span, bay width, number of stories, number of bays, or value of loads. In general, these are single valued variables. While there may be more than one story, a single story will have only one height.

Problem constraints are the limitations within which the problem requirements must be met. These may include such items as the rolled shapes available, plate thicknesses and lengths available, yield strengths available, and the unit costs of various materials. Generally the problem constraints will be multi-valued. They are usually concerned with material availability and costs.

Problem variables together with the problem constraints include all those items necessary for a complete definition of the structural design problem. Generally these are items which are specified by someone other than the structural engineer. While in some cases the engineer may have a voice in the determination of some of these variables, they are generally determined by the owner, the architect, building codes, specifications, or vendors.

Design parameters and design constraints, on the other hand, are completely subject to the control of the structural engineer. The distinction between the two is somewhat arbitrary as items may be interchanged between them. Design parameters are those items which when assigned values reduce the design problem to essentially a linear selection problem on the remaining items, which are then defined as the design constraints. A trial design is specified by giving values to both the design parameters and the design constraints.

As an example, in a plate girder design, once the depth of the girder and the width of the flanges are set, then thicknesses of the flanges and web may be obtained by an essentially linear selection from the available plate sizes. In this example the span of the girder and the loads to be carried would be classed as problem variables while available plate sizes and their costs would be classed as problem constraints. The depth of the girder and width of the flanges are taken as design parameters while the thicknesses of the web and flanges are taken as design constraints. It is obvious that the thickness of the flange and its width could be interchanged as to class.

Of importance in several of the sections that follow will be the mode of variation of the various parameters. Values of these parameters may be

1. a constant.

2. a member of an array.

3. a member of a set of numbers defined as

$$f_i = f_o + i(h) \text{ for } i = 1 \text{ to } n.$$

4. a continuous variable within specified limits.

In future references these modes of variation will be referred to as constant, array, incremental, and continuous respectively.

In order to choose between two feasible trial designs their relative merits must be compared. There are always some undesirable characteristics associated with any trial design. These might include, for example, cost or weight. The relationship between these characteristics and the design parameters is defined as the objective function. It is the purpose of the structural engineer to optimize this objective function or to minimize the undesirable characteristics. In this work the objective function is assumed to be a unimodal function, i.e., it is assumed that there are no local minimums. A more detailed discussion of this assumption and its limitations will be made in the chapter on optimization.

The goal of the structural design process may now be stated as follows: From the problem constraints select that set of design parameters with its associated design constraints which satisfies the problem requirements and optimizes the objective function. Figure 2 is a flow chart of a procedure to be followed in accomplishing this goal. Each of the steps in this procedure serves a particular function with respect to either one or two classes of the parameters or variables. Automation of the structural design process requires a detailed examina-

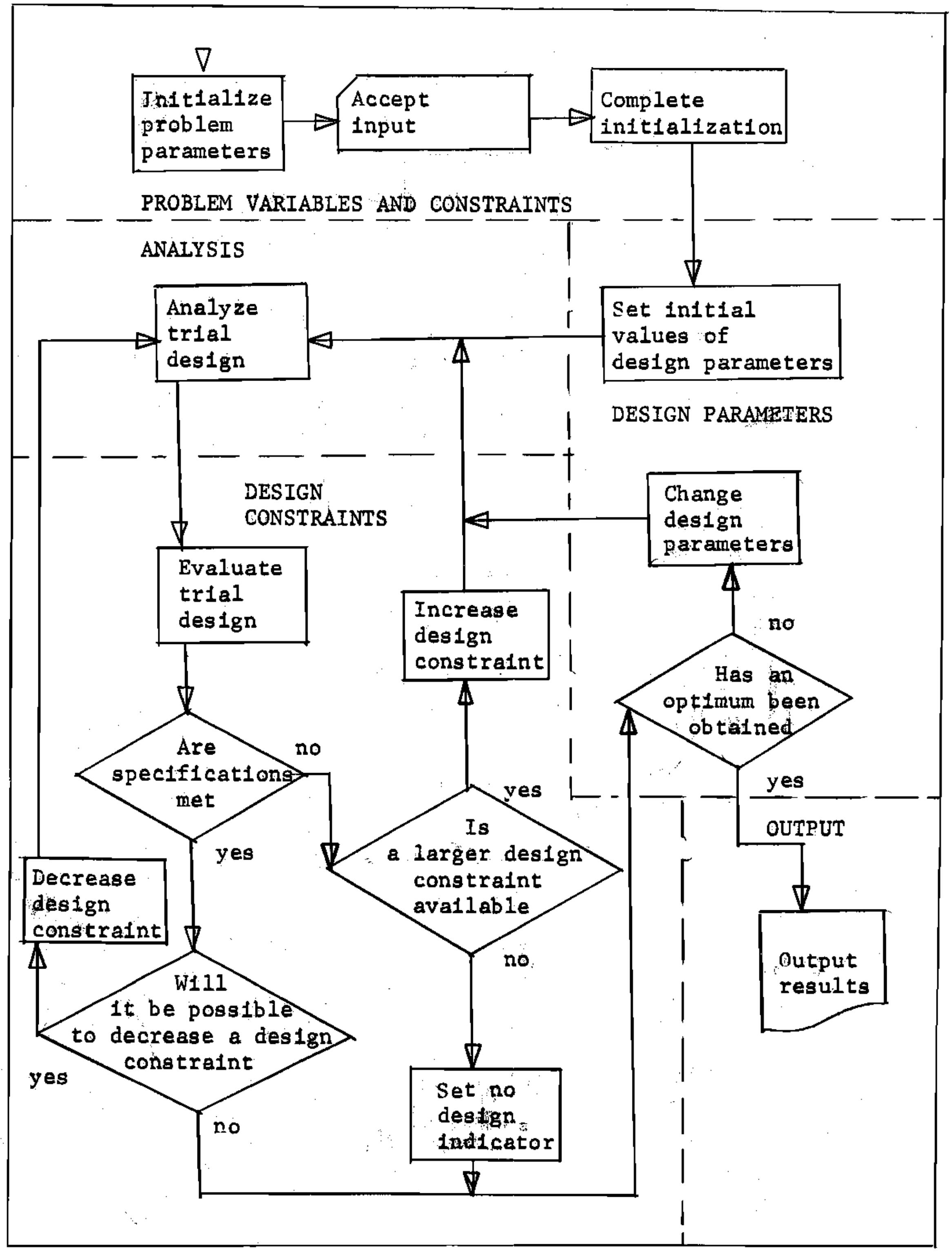


Figure 2. Expanded Structural Design Process Flow Chart

tion of the function of each step and its action on the different classes of variables. Computer routines must be capable of performing each of these functions.

Problem Variables and Constraints

The function of the initialization and input steps is to define the structural design problem; therefore, these steps deal with the problem variables and constraints. Values must be assigned to both. These values may be assigned either by default to commonly used values assigned by the initialization routine or by the user through the input routine. When at all possible the most commonly used values of the problem variables and constraints should be supplied. Certainly the user must have the ability to override these default options.

The input routine provides for user communication with the system. It should not impose rigid requirements for the form of input. The user should have complete control over the specification of the problem and the type of output. One method of eliminating the need for rigid input requirements is the use of keyword descriptive information on the data cards. This information should be in the language of the user. This together with the use of default options will allow a minimum of data input to specify a problem and will not require any fixed order of input. A convenient method of coding data is the use of a "free-form" format in which the data may be written in any column of the card with each item being delimited by a blank space. The input routine must allow for input of constant, incremental, array, and continuous data. The following examples show how this could be implemented.

1. Constant

   DEPTH OF GIRDER 36.0 INCHES

   Only one numeric value on the card indicates a constant.

2. Array

   DEPTH OF GIRDER 24.0 27.0 29.0 OR 33.0 INCHES

   More than one value on a card and the absence of the keyword "TO" indicates that this is an array of data.

3. Incremental

   DEPTH OF GIRDER 24.0 TO 36.0 BY 2.0 INCHES

   The keywords "TO" and "BY" and the presence of three numeric values indicate that this is incremental data.

4. Continuous

   DEPTH OF GIRDER 24.0 TO 36.0 INCHES

   Continuous data is indicated by two numeric values and the keyword "TO".

The input routine developed in the application portion of this work will use the arrangement outlined above.

Design Parameters

Control of the design parameters is accomplished through the use of an optimizing routine. Such a routine must allow for any of the four modes of parameter variation. The techniques used in this routine should be those which will search out the optimum values for parameters of the various modes in an efficient manner. This routine must receive the current value of the objective function at the end of each design trial. Also it must perform three primary functions with respect to the design parameters.

1. Set initial values of the design parameters.

2. Change the values of the design parameters between design trials.

3. Determine when the optimum values of the design parameters have been obtained.

A generalized optimizing routine has been developed and will be discussed in Chapter III.

Design Constraints

Values for the design constraints are selected from the problem constraints in a linear determination. The requirement on which this determination is made is that the smallest value which will satisfy the specifications should be used. Thus the function of the routines handling these variables is to determine if the specifications have been met and if so whether a smaller value would also meet the specifications. If the specifications have not been met then a larger value is used if available. If no larger value is available, then these routines must mark the particular set of design parameters as a non-feasible solution and return control to the optimizing routine for the next trial design.

New values of the design constraints are determined by moving forward or backward in a table of available values as required, forward if an increase in the value is required, backward if a decrease is required. This requires that the problem constraints from which the values are selected be in increasing sequential order. Initial values of these parameters are set by the initializing routines so that a beginning analysis may be obtained. Subsequent iterations will use the last established values as a beginning point. In statically determinate problems the changing of these values has no effect on the shears, moments, and axial forces determined by the analysis routine. Since this is not the case with statically indeterminate problems, an iterative procedure is required to insure that values are selected based on shears, axial forces, and moments which will result from these same selected values.

User's Guide Requirements

No system is complete without a clear, concise guide describing the system's functions, the basis of its operation, the methods used, the requirements for its use, and illustrative examples. Complete listings of the routines used in the system should be included as an appendix to this manual. Any machine dependent code should be noted for the convenience of those who must convert the system for use on other machines. A listing of those variable names used in the code and their meanings would be of use in modifying the routines, changing the default options, or incorporating individual routines into other systems.

Procedural Outline for the Development of a Structural Design System

1. Determine and class all variables and parameters associated with the problem. Where possible the most commonly used values of these variables should be noted for use as default options.

2. Develop a routine to initialize those values which will be provided as default options. In some cases it will be necessary to complete initialization after program input has been accepted. For

example, wind loading coefficients cannot be initialized until after the span and height of the structure have been determined. Care must be taken to insure that initialization subsequent to problem input does not override user specification. The initialization routine should also set initial values for the design constraints.

3. Develop an analysis routine to determine shears, moments, and axial forces for the different loads, and combine them in accordance with the loading combinations desired.

4. Develop an optimization routine for the functions required to handle the design parameters. The routine developed in the present work is a general one and might be used.

5. Develop routines for the functions required to handle the design constraints.

6. Write a routine to determine the value of the objective function.

7. Write a main control routine to call the various routines in the order required to attain the goal of the design process.

8. Write the User's Manual.

CHAPTER III

OPTIMIZATION

General

Optimization is the process of finding that set of parameters which produces a minimum (or maximum) value of an objective function. Optimization techniques include the methods of classical calculus, linear programming, non-linear programming, and direct searching. With the exception of the searching methods all of the above require that the objective function be stated in closed form in terms of the parameters. Optimum searching methods allow the optimization of functions whose analytical form is unknown and require only that the value of the function can be obtained by some means for any combination of the independent parameters. Since the objective function of a structural design problem usually cannot be stated directly in terms of the design parameters, optimum searching methods appear most useful and will be used in the present work.

Optimum searching methods assume that the objective function is unimodal; that is, it is assumed that no sub-optima exists. While this would appear to be a serious restriction on these methods, this obstacle may be overcome by partitioning the search area and searching each partition for its optimum. Examination of Figure 3 reveals how partitioning might be used to isolate sub-optima. Routines developed with this work will search one partition, leaving partitioning to the

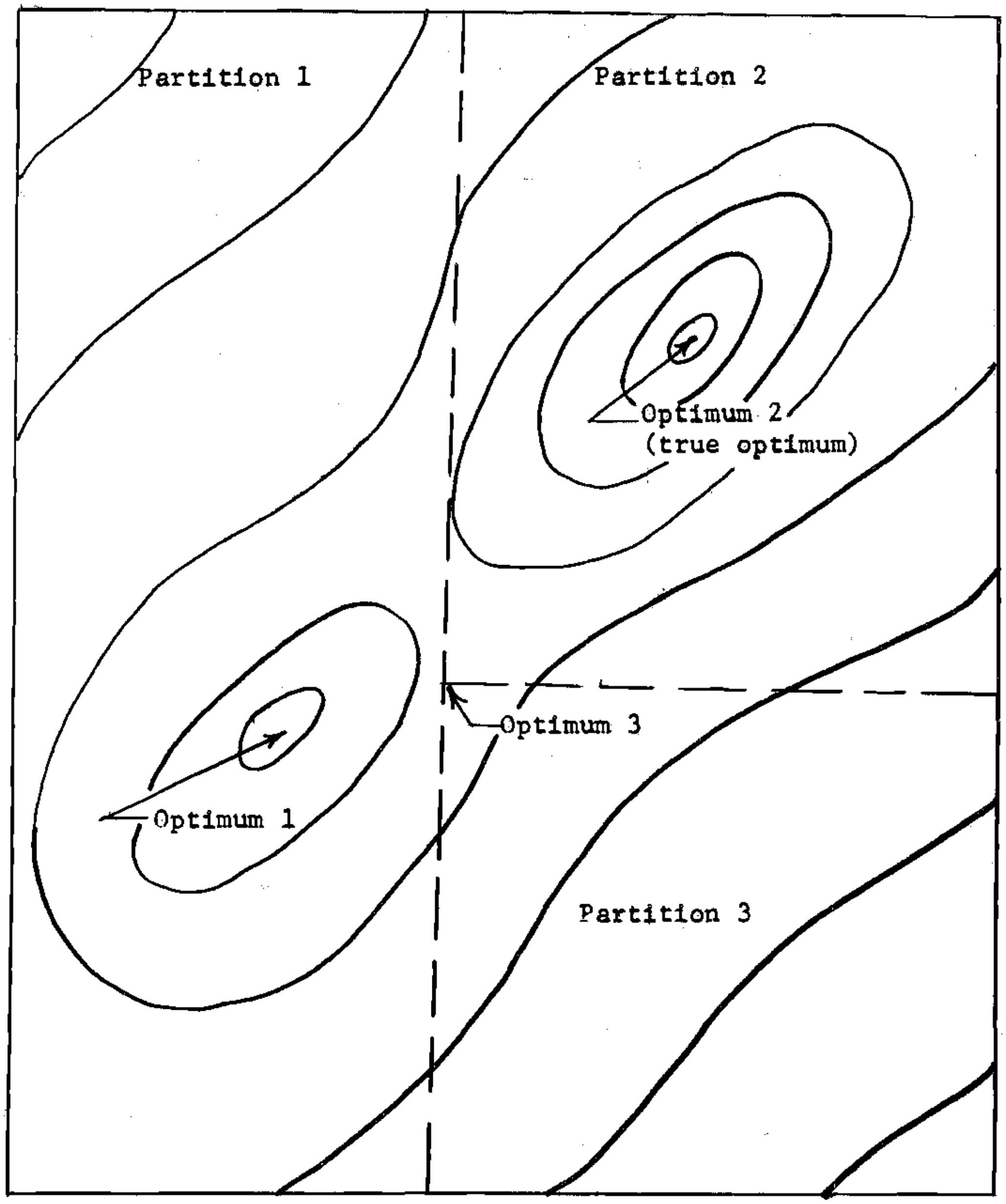


Figure 3. Partitioning to Isolate Sub-optima

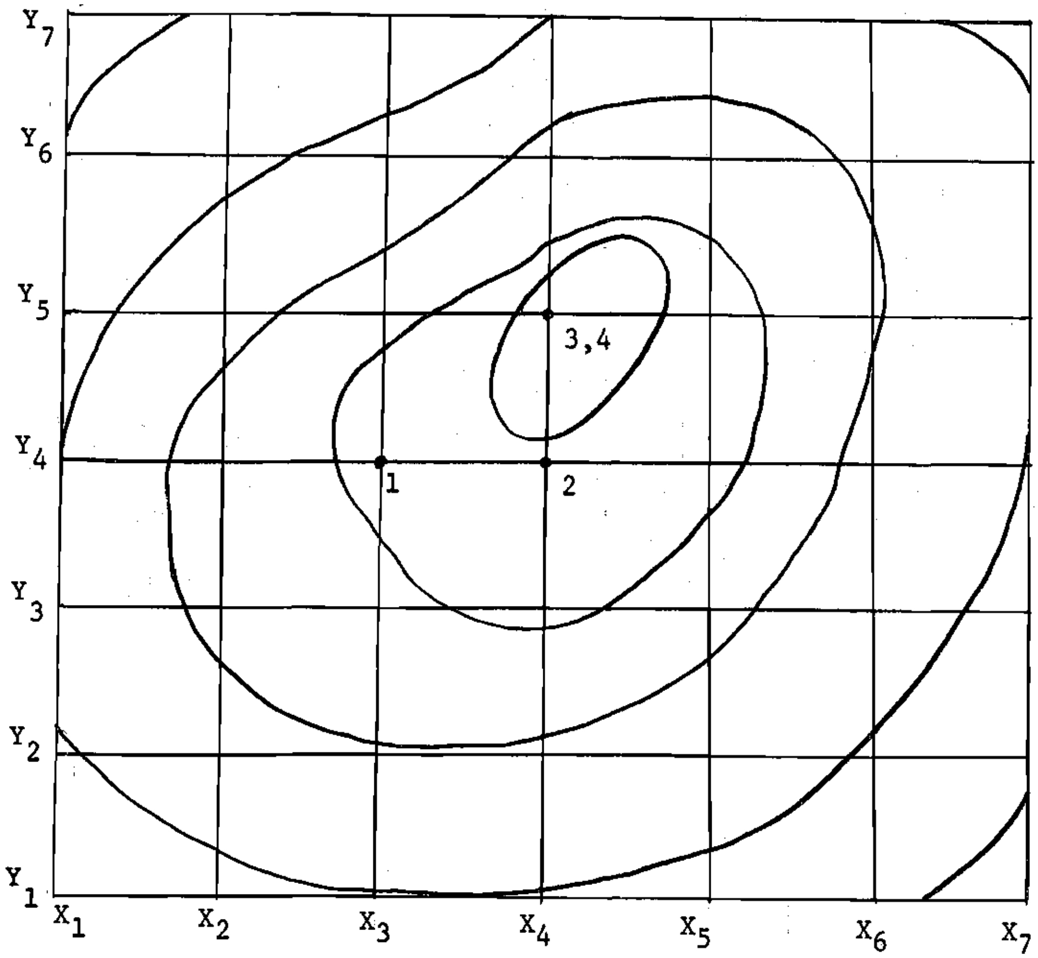
discretion of the user.

Two basic search plans will be presented in this work, namely, orthogonal and diagonal searching.

Orthogonal Searching

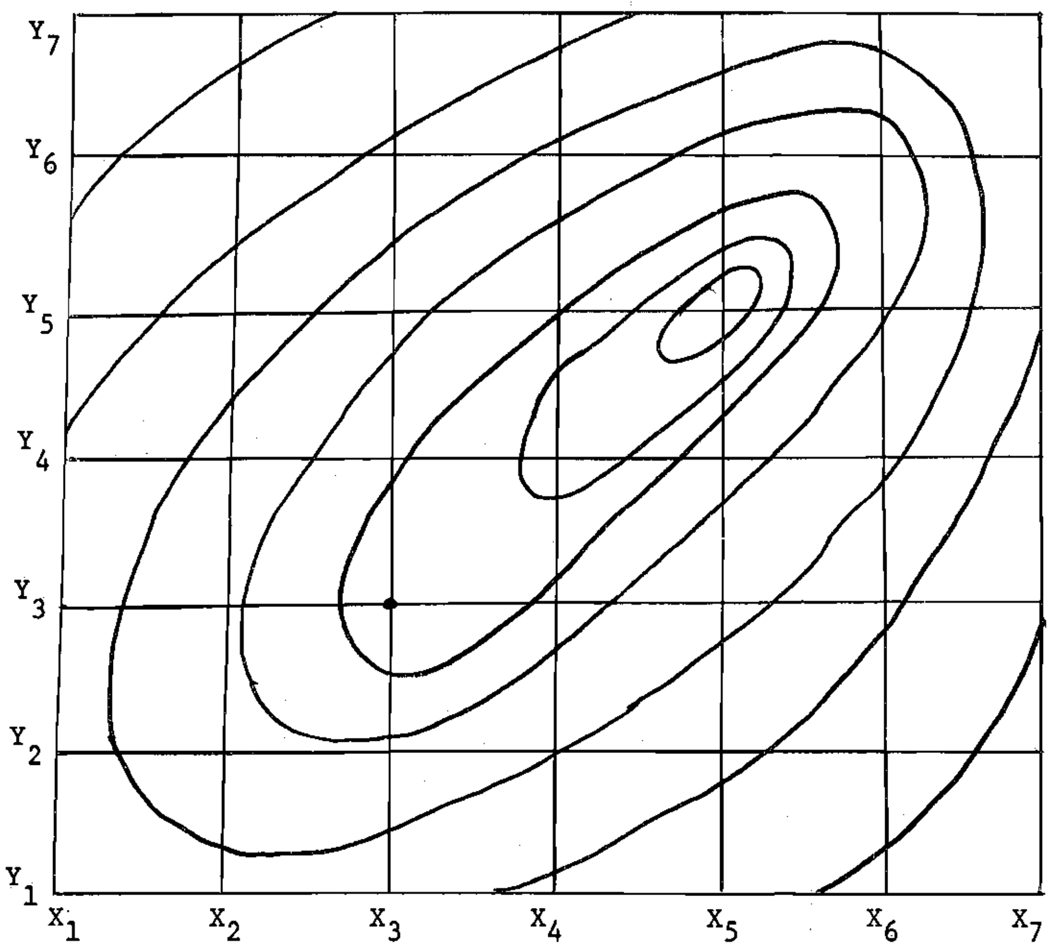
Orthogonal searching is the orthogonal application of one dimensional search techniques to multi-dimensional problems. An optimum value will be found for each parameter in turn, holding the others constant. Because of the possible interaction of the various parameters, it is necessary to repeat this process until no parameter can be changed to the advantage of the objective function. An example of this method is illustrated in Figure 4. In this two dimensional example x and y are both incremental parameters. The contours in this figure represent the values of the objective function. The first search is made with $x = x_3$, and the optimum value of y is found to be y_4. The second search is made with $y = y_4$ finding the optimum value of x to be x_4. The third search with $x = x_4$ sets $y = y_5$. The fourth search reveals that no improvement can be made in the value of the objective function by changing x. Thus the optimum has been found at $x = x_4$, $y = y_5$.

In some instances, particularly with incremental or array type parameters, orthogonal searching will indicate a false optimum (14). Such a condition is indicated in Figure 5. In this example x and y are once again incremental parameters. The first search is made with $x = x_3$, and an optimum value of y is found to be y_3. The second search finds that no improvement can be made with respect to x, thus a false



Search Order	Optimum at
1. Along Y with $X = X_3$	Y_4
2. Along X with $Y = Y_4$	X_4
3. Along Y with $X = X_4$	Y_5
4. Along X with $Y = Y_5$	X_4

Figure 4. Orthogonal Search



Search Order	Optimum at
1. Along Y with $X = X_3$	Y_3
2. Along X with $Y = Y_3$	X_3 (no change)

Figure 5. False Optimum by Orthogonal Search

optimum has been indicated at x_3, y_3. This trap will be eliminated by a non-orthogonal or "diagonal" search which will be explained later.

Search routines for structural design problems must service both continuous and array or incremental type parameters. The methods selected for use here are the golden section search for continuous parameters and a lattice search, based on the Fibonacci numbers, for array and incremental parameters. Only a brief outline of these methods will be presented here. A detailed explanation has been made by Wilde (14).

Golden Section Search

The golden section search is a method developed by Kiefer (5) based on Euclid's golden section. The golden section is the division of a line into two parts such that the ratio of the longer part, x, to the whole, L, equals the ratio of the shorter part, L - x, to the longer part.

$$\frac{x}{L} = \frac{L - x}{x}$$

$$\frac{x}{L} = 0.61804 \qquad \frac{L - x}{L} = 0.38196$$

When using this method to locate an optimum value of an objective function within some range, L, of the independent variable, the first two determinations of the function value should be made at points L - x from each end of the range. Since the function is assumed to be unimodal, the range containing the optimum may be reduced by discarding that region to the right or left of the higher function value, to the right if the right value is high, to the left if the left value is high. This process is continued until the range containing the optimum has been reduced to some acceptable value. Use of the golden section requires that only one additional function evaluation be made for each range reduction after the first. This process is illustrated in Figure 6.

Lattice Search

The lattice search is another technique developed by Kiefer (5). This search plan is useful where the independent variable cannot vary continuously but must take on discrete values. Use is made of the Fibonacci numbers which are defined as follows:

$$F_0 = F_1 = 1$$

$$F_n = F_{n-1} + F_{n-2} \qquad \text{for } n = 2,3,4,5,\ldots\ldots$$

In this plan the number of discrete values of the independent variable must be one less than a Fibonacci number. If this is not the case, dummy values must be added to meet this requirement. The two numbers whose sum produced the Fibonacci number one larger than the number of discrete values of the independent variable are taken as the indices of the first two searches. Range reduction is accomplished as with the golden section method. Once again only one function evaluation will be required for each range reduction after the first. This method is illustrated in Figure 7.

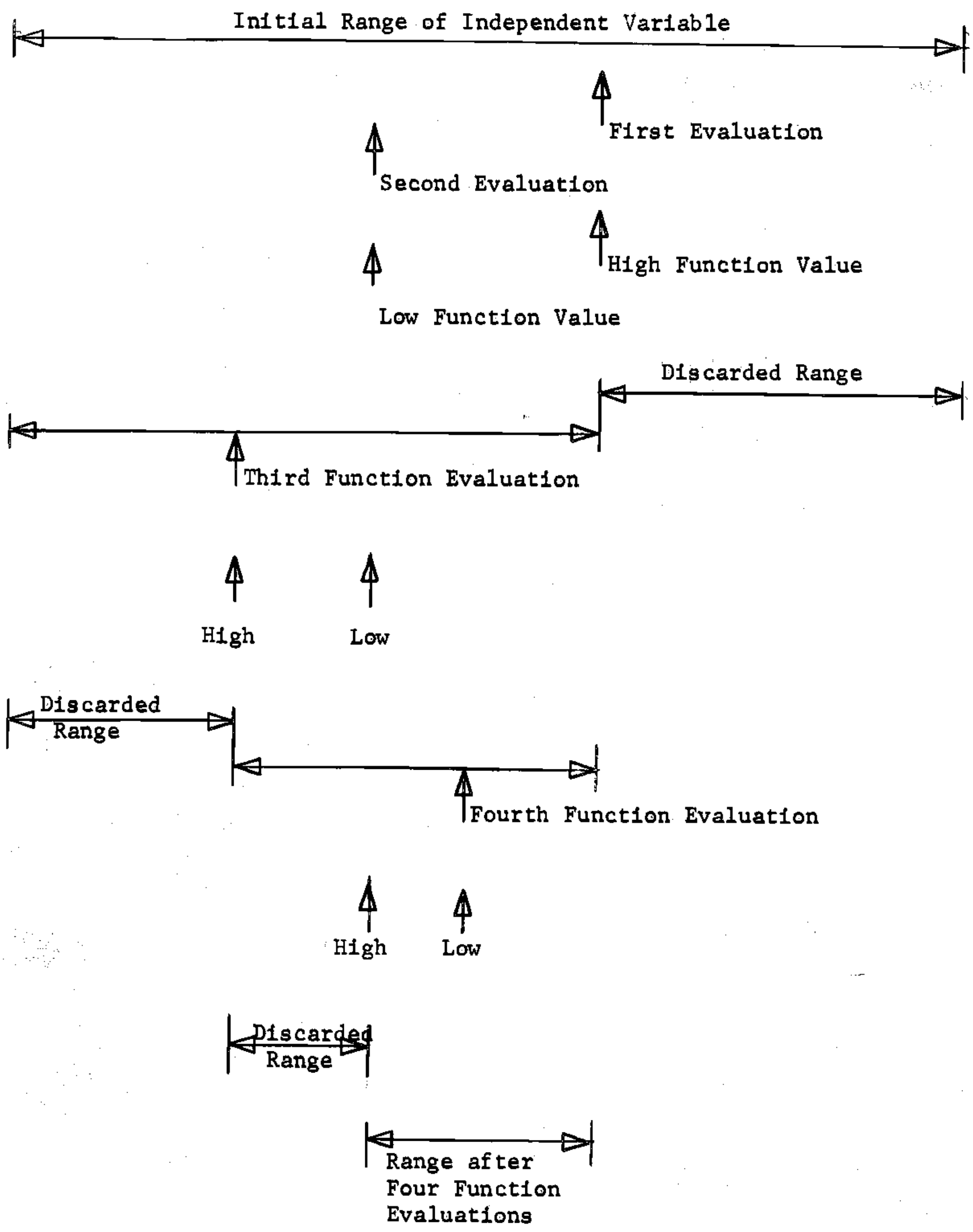


Figure 6. Golden Section Search

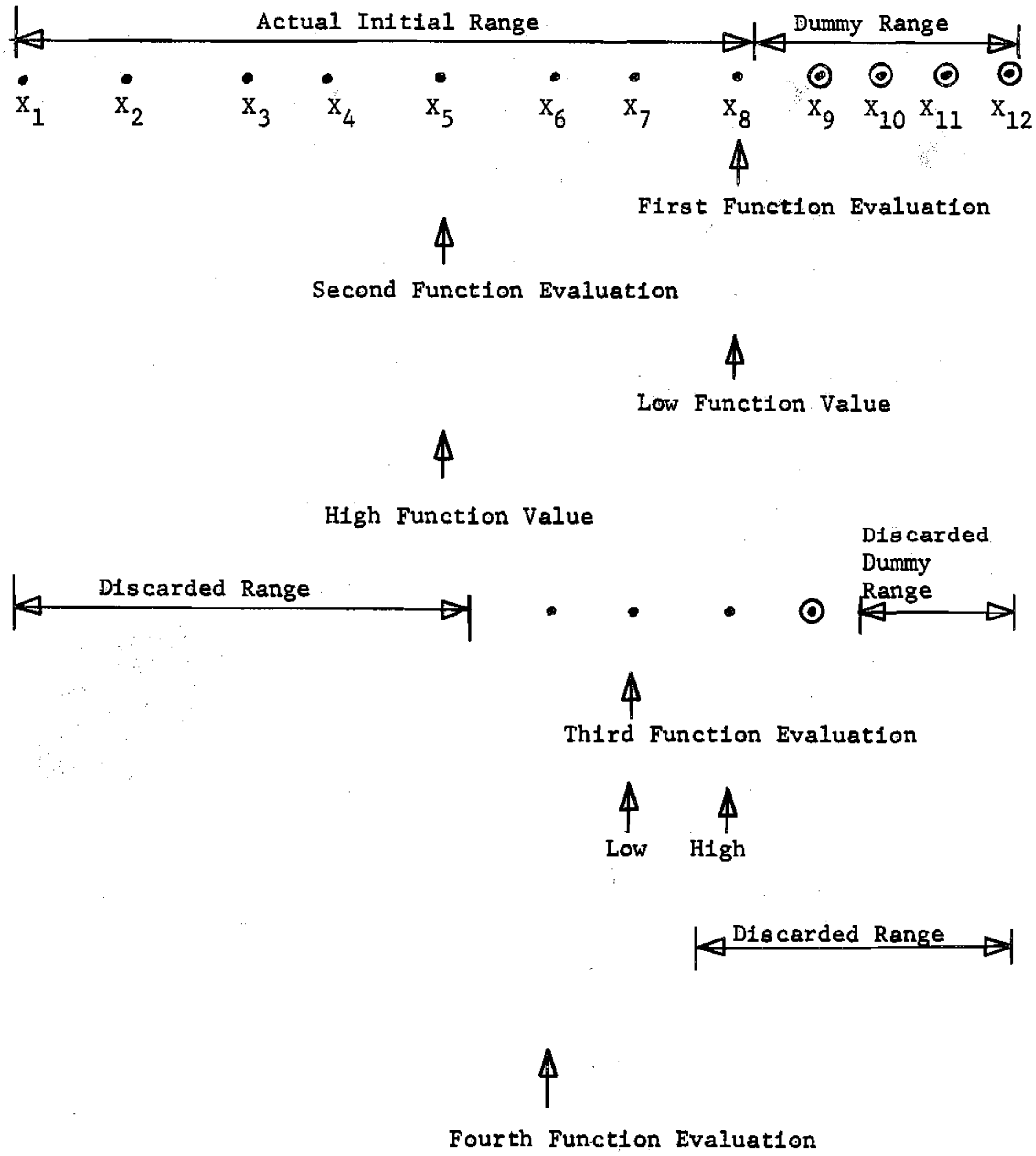


Figure 7. Lattice Search

Diagonal Searching

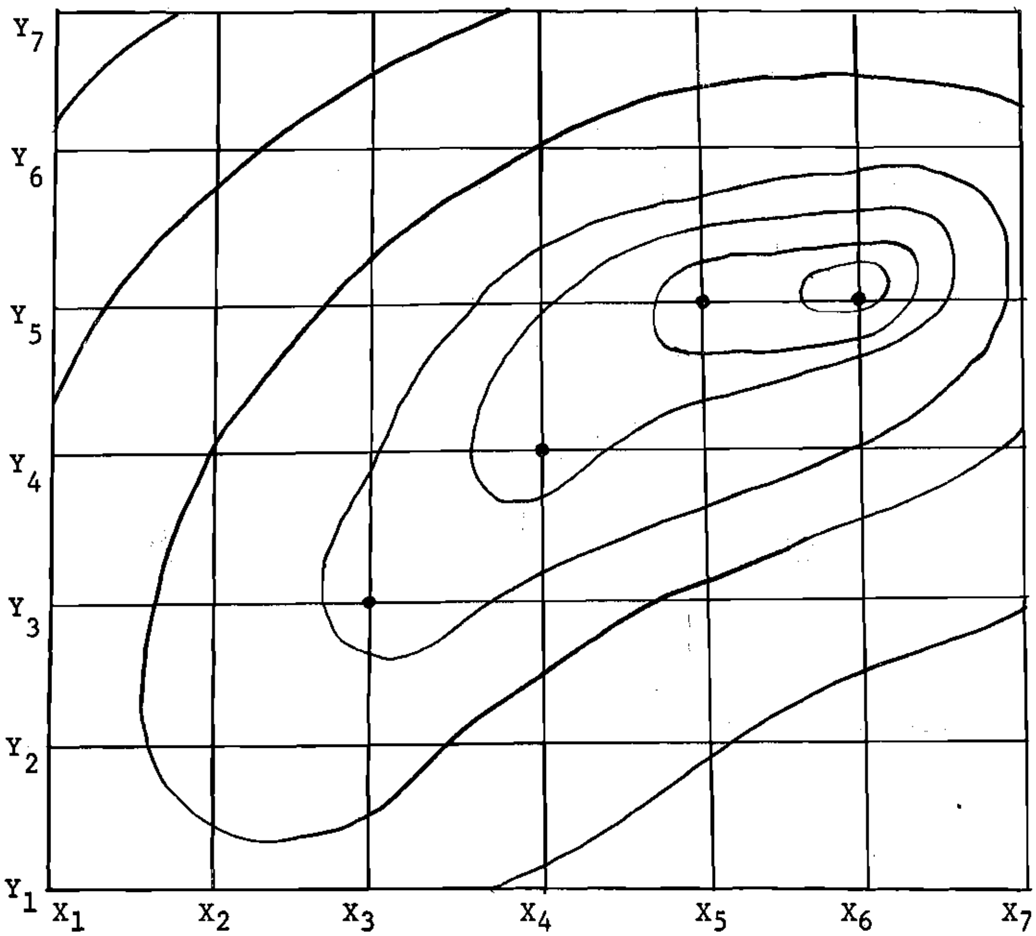
As previously shown, orthogonal searching will not necessarily produce an optimum value of the objective function. To insure that an optimum solution has been reached it will be necessary to check the points on the non-orthogonal vectors adjacent to the indicated optimum. If none of these points produces a better value of the objective function, then the optimum has been located for a unimodal function. If a better value is found, then diagonal searching continues along this vector one step at a time as long as each step produces an improvement in the objective function. Diagonal searching which produces an improvement in the objective function will be followed by orthogonal searching. Orthogonal searching is always followed by diagonal searching. Diagonal searching of points adjacent to the indicated optimum with no improvement in the objective function terminates the search, as the optimum will have been obtained. This overall process is illustrated in Figure 8.

Searching Subroutines

Two subroutines have been developed to accomplish the search plan outlined in this chapter. Because of their importance in the practical application of optimum design and because of their general applicability to other optimizing problems, their functions and requirements will be discussed in detail.

These subroutines, OPTMUM and SEARCH, are written in basic FORTRAN and may be called as follows:

CALL OPTMUM (KIN)



<u>Search Order</u>

1. Orthogonal search $X = X_3$
2. Orthogonal search $Y = Y_3$
3. Diagonal search around X_3, Y_3
4. Diagonal search of X_5, Y_5
5. Diagonal search of X_6, Y_6
6. Orthogonal search $X = X_5$
7. Orthogonal search $Y = Y_5$
8. Orthogonal search $X = X_6$
9. Diagonal search around X_6, Y_5

Figure 8. Overall Search Plan

CALL SEARCH (SWTCH)

KIN and SWTCH are indices used both by the calling program and by the subroutines to control the search plan. Data communication between the calling program and these subroutines is through the COMMON area. The calling program must contain a COMMON statement similiar to the following:

COMMON A(9,17),OPTL(6),OPSAV(10)

The tables established by this statement are illustrated in Figure 9. Prior to the use of these tables the items checked in this figure must be supplied. The other items will be supplied as necessary upon the first call to OPTMUM. It should be noted that each parameter to be optimized is assigned to a line in the A matrix. For convenience, the names of these parameters in the calling program may be set equivalent to their present values in the A matrix.

EQUIVALENCE (COLU,A(3,2)), (RAFT,A(5,2))

Subroutine OPTMUM has two primary functions: initialization of the tables required for the optimization process and the control of the orthogonal search pattern. It uses the golden section method for continuous parameters and the lattice search method for array and incremental parameters. It takes no notice of constant parameters. The initial call is made to OPTMUM with KIN = 1. At this time the A table is completed, the OPSAV and OPTL tables initialized, and control of the search given to OPTMUM. A return from OPTMUM with KIN = 2 or 3 requests the calling program to evaluate the objective function for the present parameter values indicated in the A table, store this value as the new function value in the OPTL table, and recall OPTMUM. A return from

Type of Parameter	Present Value	Lower Limit	Upper Limit	Reduced Lower Limit	Reduced Upper Limit	Increment	Array of Values									
							1	2	3	4	5	6	7	8	9	10
0 ✓	✓															
1 ✓		✓	✓				✓	✓	✓	✓	✓	✓				
2 ✓		✓	✓			✓										
3 ✓		✓	✓													

The 'A' Table

<u>Type of Parameter Codes</u>

0 – Constant

1 – Array

2 – Increment

3 – Continuous

✓ Number of Parameters	New Parameter Value	New Function Value	Present Parameter Number	Old Parameter Value	Old Function Value

The 'OPTL' Table

Optimum Parameter Values									Function Value
1	2	3	4	5	6	7	8	9	

The 'OPSAV' Table

Figure 9. Control Tables for the Optimizing Subroutines

OPTMUM to the calling program with KIN = 4 indicates that orthogonal searching has been completed.

Subroutine SEARCH controls the diagonal search pattern. Search control is given to SEARCH with SWTCH = 0.0. Return from SEARCH to the calling program with a negative value of SWTCH indicates that diagonal searching has been completed without improvement of the objective function. Return with SWTCH = 0.0 indicates that diagonal searching did improve the objective function. Return with a positive value is a request to the calling program to evaluate the objective function for the parameter values indicated in the A table, store this value in the OPTL table, and recall SEARCH.

At the completion of either diagonal or orthogonal searching the optimum values of the parameters and the associated value of the objective function are contained in the OPSAV table. The overall control of the search plan is left to the calling program in that it has control over the order in which OPTMUM and SEARCH are called. A complete listing of the FORTRAN statements for these subroutines is contained in the Appendix.

CHAPTER IV

APPLICATION

System Description

The type of structure selected to demonstrate the proposed method is a single span gabled frame constructed of built-up sections. A typical frame of this type is illustrated in Figure 10. A computer soft-ware system has been developed for the automated optimum design of this type of frame. This system is written in basic FORTRAN and has been successfully run on a 131,072 byte IBM 360 Model 40 computer. A listing of the FORTRAN source statements for this system is included in the Appendix to this paper. Since the capability of the system will become apparent with the presentation of the User's Guide, only the organization of the system will be presented here. Development of this system was accomplished following the outline presented in Chapter II.

Variable Classification

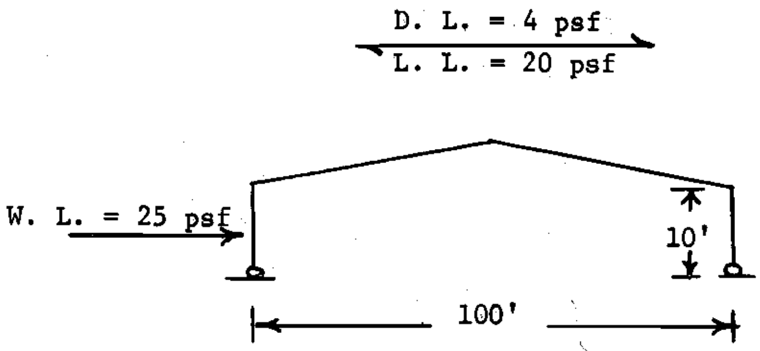
The first step in the development of an automated structural design system is the identification and classification of the parameters and variables associated with the problem. For this system the following classifications were made.

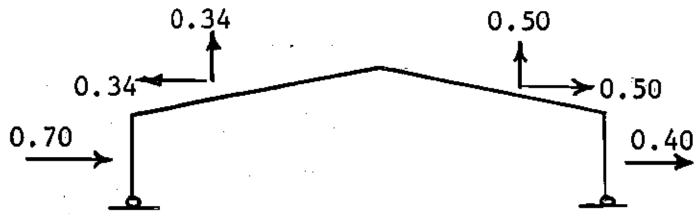
Problem Variables

(1) Span of the frame

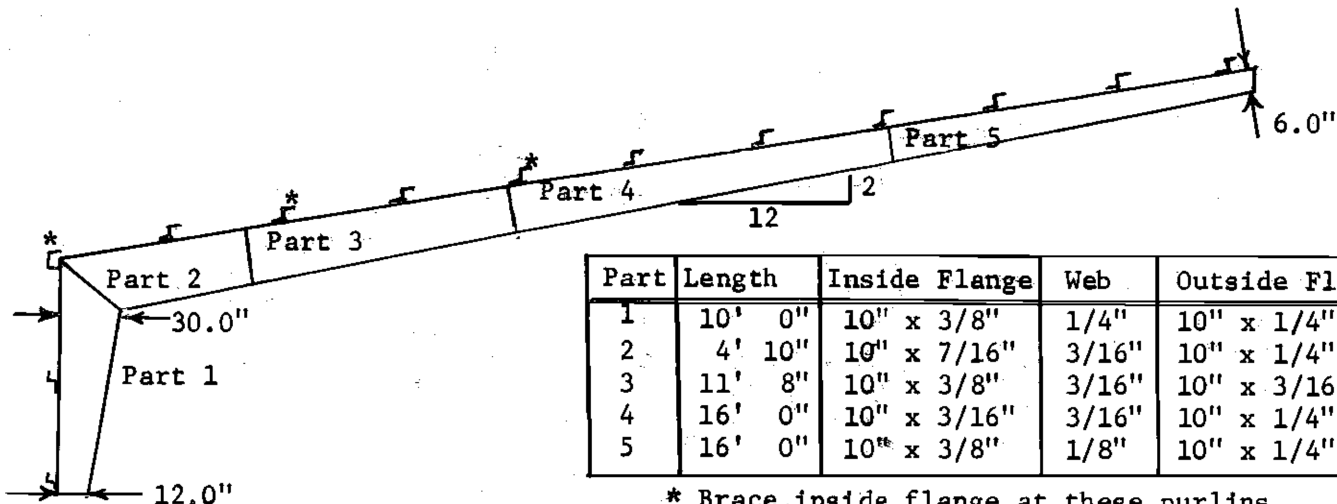
(2) Height of the frame at the eave

(3) Loads and their locations





Wind Coefficients



Part	Length	Inside Flange	Web	Outside Flange
1	10' 0"	10" x 3/8"	1/4"	10" x 1/4"
2	4' 10"	10" x 7/16"	3/16"	10" x 1/4"
3	11' 8"	10" x 3/8"	3/16"	10" x 3/16"
4	16' 0"	10" x 3/16"	3/16"	10" x 1/4"
5	16' 0"	10" x 3/8"	1/8"	10" x 1/4"

* Brace inside flange at these purlins.

Figure 10. Typical Frame

(4) Girt and purlin depths

(5) Yield strengths of the steels

(6) Costs of the steels

Problem Constraints

(1) Plate thicknesses

(2) Plate lengths

Design Parameters

(1) Length of the parts

(2) Depth of the section at the eave

(3) Slope of the rafter

(4) Depth of the section at the column base

(5) Depth of the section at the ridge

(6) Width of the rafter flanges

(7) Width of the column flanges

(8) Location of any change in geometry in the rafter

(9) Depth of the section at any change in geometry in the rafter

(10) Minimum A/H ratio for web stiffeners

Design Constraints

(1) Web thicknesses

(2) Inside and outside flange thicknesses

Initialization and Input Routines

Four subroutines were written to handle the initialization and input of the problem variables and problem constraints. The names of these subroutines along with a brief description of their functions are listed below.

NITIAL - initializes the problem variables and constraints. This routine also sets the initial values of the design constraints.

INPUT - accepts the user's data.

LOOK - scans an input record for one of a list of keywords.

SCAN - scans input records for numeric data.

WIND - computes wind loading coefficients if not specified by the user.

Analysis

In the analysis a horizontal reaction is selected as the redundant reaction and its value is computed as

$$H = \frac{\int \frac{M_s y \, ds}{EI}}{\int \frac{y^2 ds}{EI}}$$

where M_S is the moment in the simple frame with the redundant reaction removed. Integration is approximated using Simpson's Rule with seven sections per frame part. The trapezoidal rule is used to account for the excess portions of the frame at the eaves and ridge. Shears, moments, axial forces, and reactions are combined in the combinations requested for the design. The following six subroutines are used in the analysis section of the system.

GEOM - computes the x and y coordinates and depths of each section.

UNIF - computes reactions, shears, moments, and axial forces at each section for uniformly distributed loads.

CONCEN - computes reactions, shears, moments, and axial forces at each section for concentrated loads.

COMBIN - combines reactions, shears, moments, and axial forces into the requested design combinations.

UNSUL - computes unsupported lengths.

LOCATE - locates purlins and girts.

Manipulation of the Design Parameters

Manipulation of the design parameters to determine an optimum design is controlled by three subroutines. These are:

PARTS - selects an optimum combination of part lengths for column and rafter to minimize material waste.

OPTMUM - initializes design parameters and controls orthogonal searching to minimize the objective function.

SEARCH - controls diagonal searching.

Evaluation of the Design Constraints

For each trial design the minimum values of the design constraints which will meet the required specifications must be selected. The following subroutines are used for this function.

SELECT - selects minimum values of design constraints which will satisfy the American Institute of Steel Construction specifications.

SSTRES - computes the allowable shear stress at a section.

STRESS - coordinates allowable bending and axial stress computation.

BSTRES - computes allowable bending stress at a section.

FINDER - is used to determine the index of the design constraint currently being used.

CSTRES - computes allowable compressive stress.

INERT - computes moment of inertia and section moduli at all sections of a part.

Main Control Program

The main control program directs the flow of the problem by calling the various subroutines. This routine also controls the evaluation of the objective function. In this case the objective function is the unit cost of the steel times the weight of the frame plus the number of stiffeners times a fixed cost per stiffener.

Prior to listing the output any unnecessary inner flange braces are removed and required web stiffeners are located. These functions along with the formal output are handled by the main control program and the following subroutines:

STFNER - locates web stiffeners.

PRINT - prints shears, reactions, moments, and axial forces for the individual loads and for the design combinations.

OUTPUT - prints the stresses for the design combinations.

CHECK - computes the design factor for the flange brace removal routine.

The relationship between the main control program and the 25 subroutines is shown in Figure 11. Flow of the problem is through the initialization and input routines, the analysis section, the routines to select values for the design constraints, the optimizing section, looping back to the analysis section until an optimum is achieved, then finally to the output section.

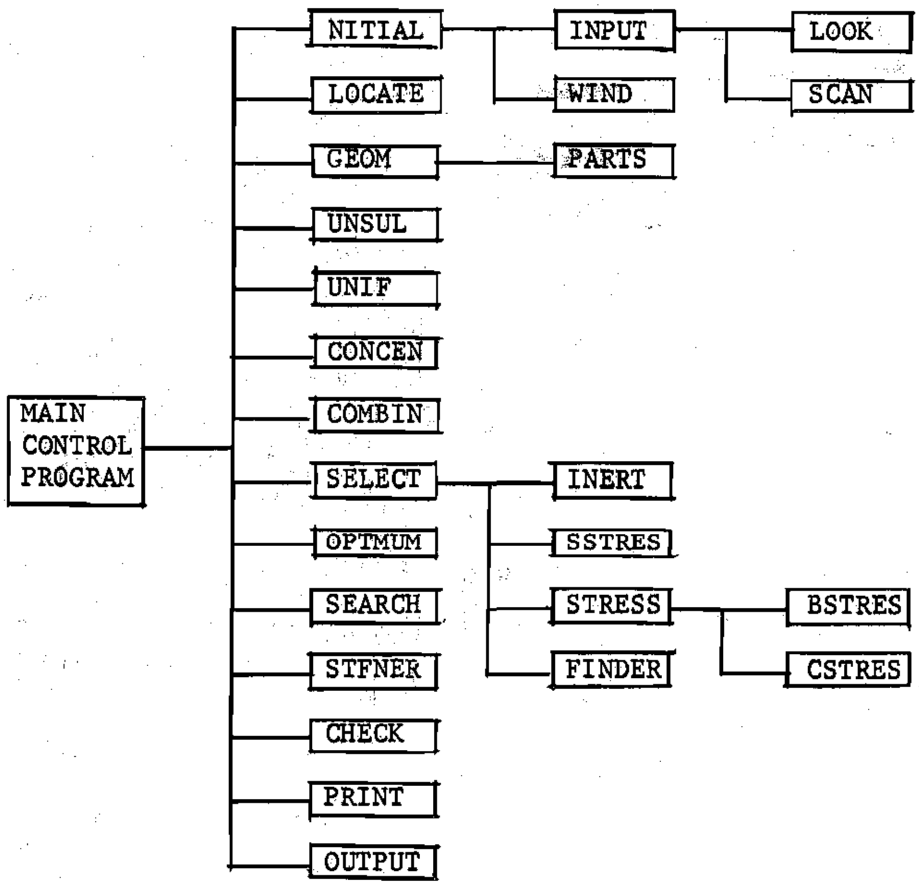


Figure 11. System Organization Chart

CHAPTER V

USER'S GUIDE

Problem Specification

This chapter will explain how a user specifies his problem to the system described in Chapter IV. An attempt has been made to make this specification as simple as practicable yet provide the user with complete control over the action of the system. Thirty-one control cards are provided by the system for problem specification. Only three cards are absolutely necessary to the system. These are the SPAN, HEIGHT, and EXECUTE cards. Where cards are omitted the system will supply commonly used values for the variables. Since all such default action is taken in subroutine NITIAL, it would be a simple matter for a user to change the default values provided by the system.

Each of the 31 control cards is identified by a keyword. This keyword must begin in column one of the card and at least the first four characters supplied. The entire word or phrase may be supplied if desired, and comments may be written on the card provided they contain no numeric characters. The required numeric input data may be written in any column after the fourth. Each value is delimited by a non-numeric character or blank space. For this purpose the plus sign, the minus sign, and the decimal point are considered numeric characters. While the data may appear in any column of the card, it must be in the order specified for that type of card.

The type of parameter variation is specified on the card along with the data. Constants are indicated by the presence of only one numeric value on a card. Arrays of data are indicated by the presence of two or more numeric values on the card and the absence of the keyword "TO". Incremental variation is indicated by three numeric values and the keywords "TO" and "BY". Continuous variation is indicated by two numeric values and the keyword "TO". Examples of each of these will be used in the explanation of the control cards.

With two exceptions, there is no specific or required order for the control cards. When a PARTS card is used, then it must be immediately followed by one descriptive card for each part of the frame. The other exception is the EXECUTE card which must be the last input card for the problem.

An example problem illustrating both user input and system output is included in the Appendix.

Control Cards

The general form of each of the 31 control cards along with their options, units, and default values will be presented in the alphabetical order of the keywords.

ALTERNATE STEEL 42000.0 PSI COST 0.16 DOLLARS

This card specifies the yield strength and cost of a second steel. It is used in conjunction with the VARY card. There is no default option available for this type card.

ANALYZE ONLY

An ANALYZE card specifies that only an analysis is desired.

Analysis output should be requested or defaulted. The frame to be analyzed must be completely described using PARTS cards. If the system is to design the frame, this card is omitted.

AOH 2.0 TO 3.0 RATIO

This card specifies the minimum A/H ratio the system may use for web stiffeners. The data may be constant, array, incremental or continuous. Omitting this card specifies that web stiffeners are not to be used.

BASE DEPTH 7.0 9.0 11.0 INCHES

The depth of the frame at the column base is described by this card. The data is in inches and may be constant, array, incremental or continuous. The default option if this card is omitted is shown in the example shown above.

BAY 20.0 FEET

This card indicates the spacing of the frames in feet. This is a problem variable and is constant. If this card is omitted, 20 feet is assumed.

BRACE AT 20.0 AND 30.0 FEET

This card specifies the horizontal distances from the center of the column base at which inside flange braces will be placed for purposes of erection. The distances may be approximate as the system will place the brace on the closest purlin. It is assumed that these braces will be left in place and are suitable for lateral support of the inside flange. This card may be omitted if not applicable. The default option specifies that no erection braces are left in place. The data supplied with this card must be a constant or an array.

COLUMN WIDTH 5.0 6.0 7.5 9.0 10.0 12.0 INCHES

The data for the column width is supplied by this card. It may be constant, incremental, array, or continuous. In the event this card is omitted the array shown in the example above is assumed.

COMBINATION 1.0 1.0 0.0 0.75 0.0 0.0 0.0 RATIO

Up to four of these cards may be used to specify the load combinations desired for the design. Seven numbers must be provided on each such card. These numbers represent respectively the portion of each of the following loads to be included in the particular design combination.

1. Dead load
2. Live load
3. Wind load
4. Concentrated load number one
5. Concentrated load number two
6. Concentrated load number three
7. Concentrated load number four

COST OF STIFFENERS 2.00 DOLLARS

This card indicates the cost of web stiffeners. If this card is not supplied a value of zero is assumed. Data supplied must be a constant.

DBREAK 20.0 TO 36.0 BY 2.0 INCHES

Data for the depth of the frame at any change in geometry in the rafter is supplied by this type card. It may be incremental, constant, array, or continuous. This card must be supplied only if an XBREAK card is supplied.

DEPTH OF GIRT 7.5 INCHES

The depth of the girts to be used is indicated to the system as shown above. This is a problem variable and thus is a constant. Its value defaults to 7.5 inches if this card is not supplied.

EAVE DEPTH 36.0 TO 48.0 INCHES

This card supplies the data for the depth of the frame at the eave. This data may be constant, array, incremental, or continuous. Omission of this card causes the system to provide continuous data with limits of two and one half per cent to three and one third per cent of the span.

EXECUTE

The EXECUTE card indicates that all the problem data has been supplied. This card must be the last card of the problem input and cannot be defaulted.

FLANGE THICKNESSES .1875 .25 .3125 .375 .4375 .5 .5625 .625 INCHES

The available flange thicknesses are supplied by a card of this type. The data must be an array. The values supplied in the event this card is omitted are those shown in the example above.

GIRT LOCATION 0.5 3.0 8.0 12.0 16.0 FEET

The heights of the girts above the base of the column are indicated with this card. If this card is omitted girts are assumed at one half foot, three feet, eight feet, and approximately every four feet above eight feet. The data supplied must be an array. The eave strut is counted as both a purlin and a girt.

HEIGHT OF THE EAVE 20.0 FEET

The height of the eave measured in feet along the sidewall sheeting is specified as shown above. This card must contain a constant and must be supplied. It cannot be defaulted.

LOAD

There are several types of LOAD cards. Each will be discussed individually and each may be individually defaulted.

LOAD DEAD 4.0 POUNDS PER SQUARE FOOT

The dead load is indicated with this type of LOAD card. The data must be a constant and has a default of four pounds per square foot. The keyword DEAD must appear on the card but in no particular column. It does not have to preceed the value of the load.

LOAD LIVE 20.0 POUNDS PER SQUARE FOOT

The live load is indicated to the system in the same manner as the dead load but with the keyword LIVE. The default value for the live load is 20 pounds per square foot.

LOAD WIND

One of four wind load cards may be supplied depending on the type of wind load application desired. If no wind load card is supplied then the system assumes a wind load of 25 pounds per square foot applied in accordance with the specifications of the Metal Building Manufacturer's Association.

LOAD WIND ASCE

This card specifies that the wind load and method of application are to be in accordance with the recommendations of the American Society of Civil Engineers.

LOAD WIND PROJECTION 20.0 POUNDS PER SQUARE FOOT

Supplying this card causes the system to apply a wind force of the indicated magnitude to the vertical projection of the building.

LOAD WIND MBMA 20.0 POUNDS PER SQUARE FOOT

A wind load card with the keyword MBMA causes a wind load of the indicated magnitude to be applied in accordance with the specifications of the Metal Building Manufacturer's Association.

LOAD WIND ARRAY 20.0 PSF 0.70 -0.34 -0.34 -0.50 -0.50 -0.40

The keyword ARRAY on the wind load card indicates that the six wind load coefficients are supplied on the card following the magnitude of the load. These coefficients are supplied in the following order with a minus sign indicating suction.

1. Horizontal coefficient for windward wall
2. Horizontal coefficient for windward roof
3. Vertical coefficient for windward roof
4. Vertical coefficient for leeward roof

5. Horizontal coefficient for leeward roof

6. Horizontal coefficient for leeward wall

LOAD CONCENTRATED 2 20.0 0.0 5000.0 0.0 0.0 RIGHT RAFTER

LOAD CONCENTRATED 4 0.0 5.0 0.0 2000.0 0.0 LEFT COLUMN

A maximum of four concentrated loads may be specified. Each load may have a horizontal and vertical component and a moment. The data is supplied on the card in the following order.

1. Concentrated load number

2. Horizontal distance from column base to point of load

3. Vertical distance from column base to point of load

4. Vertical component of load in pounds

5. Horizontal component of load in pounds

6. Applied moment in foot pounds

RIGHT or LEFT COLUMN or RAFTER must be specified according to where the load is to be applied. The system applies all loads at the neutral axis of the section. If either item two or three above is zero the load will be shifted vertically or horizontally to the neutral axis without change in the moment. If both the horizontal and vertical location of the load are specified the load will be shifted to the neutral axis with appropriate change in the moment.

OMIT DIAGONAL SEARCHING

Inclusion of this card will cause the system to terminate searching after completing the orthogonal search. No diagonal searching would be performed.

OUTPUT PATH ANALYSIS DESIGN

The OUTPUT card allows the user to select the type output listings he desires. One or all of the keywords PATH, ANALYSIS, or DESIGN may be specified. PATH will cause the value of the objective function along with the values of the design parameters to be printed for each trial design. ANALYSIS causes the system to list the reactions, shears, axial forces, and moments for each load and each combination of load. The stresses for each combination of load and the allowable stresses will also be listed. DESIGN causes the frame description to be listed. If this card is omitted, ANALYSIS and DESIGN are assumed.

PARTS 2 4 1

This card along with associated cards which must follow it are used to describe a frame to the system for analysis. The data on the card indicates:

1. Number of frame parts in the column.

2. Number of frame parts in the entire rafter.

3. Number of frame parts in the rafter between the column and a change in geometry of the rafter.

One part description card for each part must immediately follow this card. Part description cards have the following form.

11.0 0.375 0.1875 0.3125

These cards have no keywords, and the array of data must be supplied in the following order.

1. Length of the part in feet.

2. Thickness of outside flange in inches.

3. Thickness of web in inches.

4. Thickness of inside flange in inches.

PEAK DEPTH 6.0 INCHES

This card specifies the data for the depth of the frame at the peak or ridge. This data may be constant, array, incremental, or continuous. For frames with slopes of one and a half in 12 or larger the default is a constant of six inches. For frames with slopes smaller than one and a half in 12 the default for this card is the same as for the EAVE card.

PLATE LENGTHS 16.0 11.0 8.0 FEET

The plate lengths available are indicated to the system with this card. The data is an array of one to three values with the largest listed first. The default values are the same as those shown in the example above.

PURLIN 11.0 60.0 INCHES

Data from this card is used to locate the purlins. The first value specifies the distance along the outer flange from the center of the frame to the first purlin. The second value specifies the basic horizontal spacing between the purlins. The default values for this card are the same as shown above.

RAFTER WIDTH 5.0 6.0 7.5 9.0 10.0 12.0 INCHES

The data for the rafter width is supplied by this card. It may be constant, incremental, array, or continuous. In the event this card is omitted the array shown in the example above is assumed.

SLOPE IS 2.0 IN TWELVE

The slope of the frame is indicated with this card. While this will usually be a constant, the system will allow any type data. If this card is omitted a slope of two in 12 is assumed.

SPAN 120.0 FEET

This card must be included in the input of every problem. It indicates the span of the frame measured in feet from sidewall sheet to sidewall sheet. This card cannot be defaulted.

STEEL YIELD STRENGTH 36000.0 PSI COST 0.15 DOLLARS

This card specifies the yield strength and cost per pound of the primary steel used in the design. The default values for this card are 36,000 pounds per square inch and 16 cents per pound.

TITLE PROBLEM NUMBER 3 2/11/68

This card allows the user to specify a title to be printed on each page of his output. If this card is omitted, no title will be printed. The 75 characters in columns six through 80 will be reproduced at the top of each page. This card may contain alphabetic, numeric, and special characters.

VARY 2 5 6

This card lists the numbers of the parts which are to be fabricated with the alternate steel. If this card is supplied, then the ALTER card must also be supplied. If this card is omitted, only one steel will be used in the design.

WEB THICKNESSES 0.125 0.1875 0.25 0.3125 0.375 INCHES

The available web thicknesses are supplied by a card of this type. The data must be an array. The values supplied in the event this card is omitted are those shown in the above example.

XBREAK 20.0 TO 26.0 FEET

The horizontal distance from the center of the column base to any change in geometry in the rafter is specified with this card. Only one card of this type may be supplied as the system allows only one change in geometry of the rafter. If this card is supplied, a DBREAK card must also be supplied. Omission of this card indicates a straight taper is to be used for the rafter.

CHAPTER VI

DISCUSSION

A procedure has been presented for the development of automated optimum design systems. The basis of the procedure is the classification of the variables and parameters according to who or what controls their values; the owner, the vendors, the engineer, or the specifications. Computer routines can be developed for each class. These routines together with an analysis routine may be organized as shown in Figure 2 to create an automated optimum structural design system. How well such a procedure works can be measured only in the performance of the systems thus developed. This procedure has been used to develop an automated optimum design system for variable section, single span steel frames. The resulting system has been described in the previous chapters and the appendix. This system was developed following the procedure outlined in Chapter II and then was used in its own improvement. For example, when the system was first developed only orthogonal searching capability was included. Experience with the system indicated that this search technique does not guarantee an optimum solution; therefore, diagonal searching was added. It was found, however, that diagonal searching was considerably slower than orthogonal searching and seldom produced an appreciable improvement in the objective function. Therefore, the system was modified to give the user the option of including or omitting diagonal searching.

The input and initialization routines for this system were designed to give the user complete control over problem specification, the mode and range of variation of the design parameters, and the type of output desired. All input to the system is in "free-form" format with each card being identified by a keyword. This relieves the user of the necessity of placing his data in particular columns of the cards and the cards in a particular order.

The success of optimum design systems depends upon the performance of the optimizing routines. Use of optimum searching techniques implies that the objective function is unimodal. The system developed in this work has been used to examine the variation of the objective function with respect to each of the design parameters. The form of this variation was found to be the same with respect to each of the parameters. Figure 12 is a plot of the relationship between the value of the objective function and one of these parameters, the depth of the frame at the eave. It is apparent from this figure that discontinuities and local minima often occur in the objective function due to the discrete changes in the design constraints as the design parameters are varied. While the objective function is not a smooth convex surface, the search techniques used allow the function to be examined only at discrete intervals. When examined in this fashion, the function generally appears to be unimodal. Should there be any doubt as to whether the system has found the optimum with respect to some parameter, it is a simple matter to search that parameter over whatever additional range might be desired.

A number of problems have been run on the system. The system

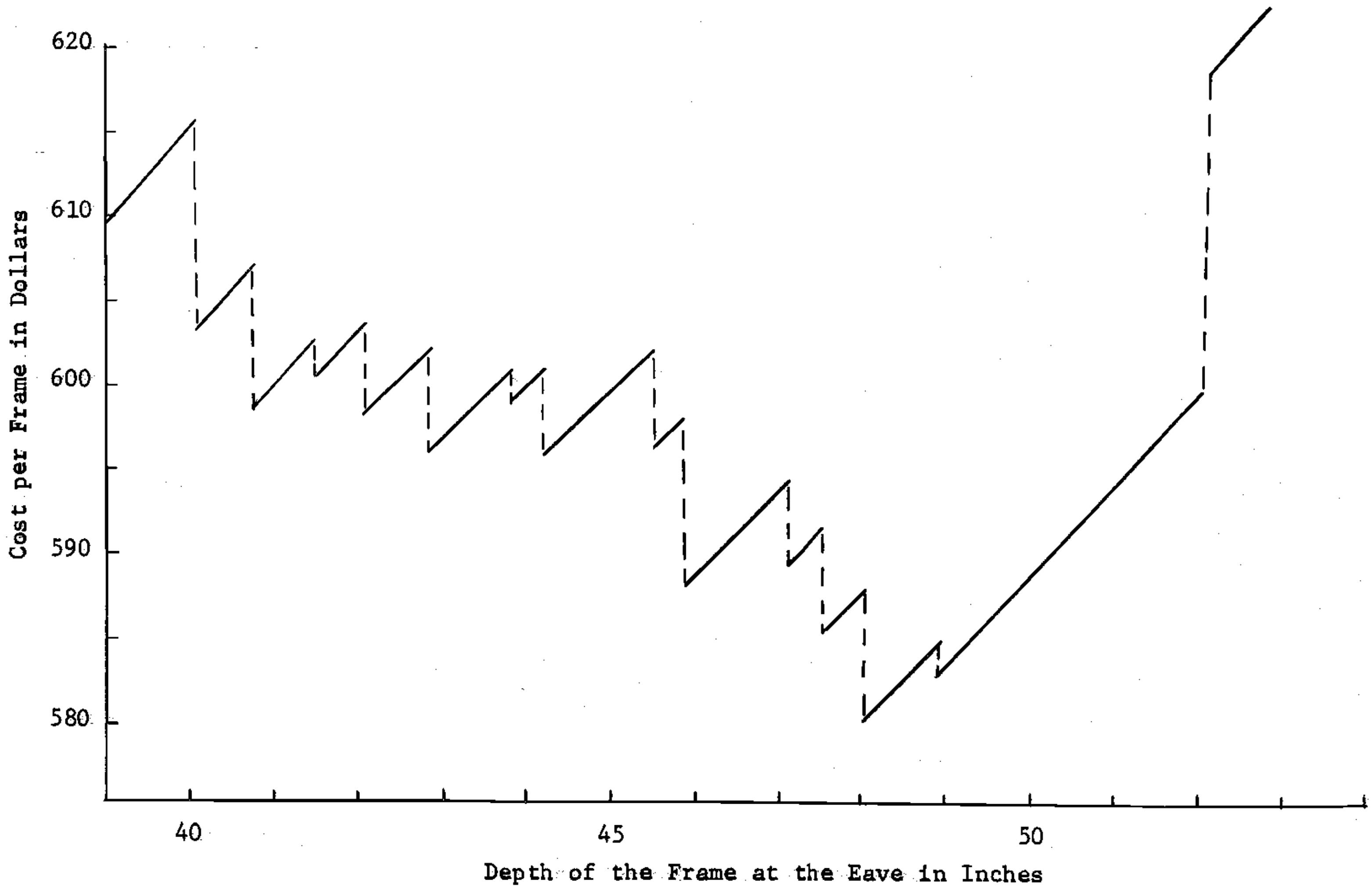


Figure 12. Partial Variation of the Objective Function

was found to be easy to control and convenient to use. In no case has difficulty been encountered in the system's obtaining a design.

CHAPTER VII

CONCLUSIONS

1. Variables and parameters of structural design problems can be classified according to who or what controls their values and computer routines written for each classification.

2. These routines together with an analysis routine can be organized according to the outline presented in this work to create automated optimum structural design systems.

3. The user of such systems can be provided with complete control over problem specification, the mode and range of variation of the design parameters, and the form of output.

4. The use of keywords and "free-form" relieves the user of the necessity of placing his data in particular columns of a card or the cards in a particular order.

5. Discontinuities and local minima may occur in the objective function of structural design problems due to the discrete changes in the design constraints as the design parameters are varied.

6. While the objective functions for structural design problems may not be unimodal, they will generally demonstrate a sufficiently convex trend to allow successful use of optimum searching techniques.

7. A successful system has been developed following the outline presented in this work for the automated optimum design of single span, variable section steel frames.

CHAPTER VIII

RECOMMENDATIONS

1. The utilization of electronic computers by engineers is in its infancy. Two current methods utilized by engineers are computer-aided design and automated design. There is merit in both approaches, and in the future some effort should be expended in combining the best of the two. For example, STRESS or STRUDL would be greatly improved if they could handle problem and design constraints, and design parameter variation to obtain an optimum design.

2. These systems, both computer-aided and automated, should not stop with just the structural design of the main frame. They should be expanded to include such items as the design of connections, bracing, and secondary framing, and the use of graphic devices for aiding in the preparation of design drawings and plans.

3. Systems such as the one developed in this work should be used to conduct parameter studies to generalize the area of optimum solutions in terms of the problem variables.

APPENDIX

EXAMPLE PROBLEM INPUT DATA

```
ALTERNATE STEEL   42000.0   PSI   0.10  DOLLARS
AOH   1.0   TO   4.0   BY   1.0
COLUMN WIDTH  5.0   6.0   7.5
COMBINATION   1.0   1.0   0.0   1.0   1.0   0.0   0.0
COMBINATION   0.75   0.0   0.75   0.75   0.75   0.0   0.0
COST OF STIFFENERS   2.00   DOLLARS
EAVE DEPTH   18.0   TO   24.0   BY   0.5
HEIGHT   14.0   FEET
LOAD CONCENTRATED 1 10.0 0.0 5000.0 0.0 0.0 RIGHT RAFTER
LOAD CONCENTRATED 2 10.0 0.0 5000.0 0.0 0.0 LEFT RAFTER
OUTPUT PATH ANALYSIS DESIGN
RAFTER WIDTH 5.0   6.0   7.5
SPAN   60.0   FEET
STEEL YIELD STRENGTH  36000.0   PSI   COST   0.08  DOLLARS
TITLE   EXAMPLE PROBLEM
VARY   1  2
EXECUTE
```

EXAMPLE PROBLEM OUTPUT

EAVE	SLOPE	BASE	PEAK	RAFT	COLU	XBRE	DBRE	A/H	COST
EXAMPLE PROBLEM									
ORTHOGONAL SEARCH									
24.00	2.00	11.00	6.00	7.50	7.50	0.0	0.0	3.00	173.79
21.50	2.00	11.00	6.00	7.50	7.50	0.0	0.0	3.00	190.18
24.00	2.00	11.00	6.00	7.50	7.50	0.0	0.0	3.00	173.79
22.50	2.00	11.00	6.00	7.50	7.50	0.0	0.0	3.00	182.99
23.50	2.00	11.00	6.00	7.50	7.50	0.0	0.0	3.00	175.86
24.00	2.00	11.00	6.00	7.50	7.50	0.0	0.0	3.00	173.79
24.00	2.00	11.00	6.00	7.50	7.50	0.0	0.0	3.00	173.79
24.00	2.00	9.00	6.00	7.50	7.50	0.0	0.0	3.00	173.15
24.00	2.00	7.00	6.00	7.50	7.50	0.0	0.0	3.00	171.67
24.00	2.00	7.00	6.00	7.50	7.50	0.0	0.0	3.00	171.67
24.00	2.00	7.00	6.00	6.00	7.50	0.0	0.0	3.00	166.49
24.00	2.00	7.00	6.00	5.00	7.50	0.0	0.0	3.00	163.03
24.00	2.00	7.00	6.00	5.00	7.50	0.0	0.0	3.00	163.03
24.00	2.00	7.00	6.00	5.00	6.00	0.0	0.0	3.00	160.02
24.00	2.00	7.00	6.00	5.00	5.00	0.0	0.0	3.00	160.52
24.00	2.00	7.00	6.00	5.00	6.00	0.0	0.0	3.00	160.02
24.00	2.00	7.00	6.00	5.00	6.00	0.0	0.0	2.00	99999.94
24.00	2.00	7.00	6.00	5.00	6.00	0.0	0.0	4.00	160.02
24.00	2.00	7.00	6.00	5.00	6.00	0.0	0.0	4.00	160.02
21.50	2.00	7.00	6.00	5.00	6.00	0.0	0.0	4.00	99999.94
24.00	2.00	7.00	6.00	5.00	6.00	0.0	0.0	4.00	160.02
22.50	2.00	7.00	6.00	5.00	6.00	0.0	0.0	4.00	99999.94
23.50	2.00	7.00	6.00	5.00	6.00	0.0	0.0	4.00	165.47
24.00	2.00	7.00	6.00	5.00	6.00	0.0	0.0	4.00	160.02
24.00	2.00	11.00	6.00	5.00	6.00	0.0	0.0	4.00	167.85
24.00	2.00	9.00	6.00	5.00	6.00	0.0	0.0	4.00	163.37
24.00	2.00	7.00	6.00	5.00	6.00	0.0	0.0	4.00	160.02
24.00	2.00	7.00	6.00	7.50	6.00	0.0	0.0	4.00	168.66
24.00	2.00	7.00	6.00	6.00	6.00	0.0	0.0	4.00	163.48
24.00	2.00	7.00	6.00	5.00	6.00	0.0	0.0	4.00	160.02
24.00	2.00	7.00	6.00	5.00	7.50	0.0	0.0	4.00	163.03
24.00	2.00	7.00	6.00	5.00	6.00	0.0	0.0	4.00	160.02
24.00	2.00	7.00	6.00	5.00	5.00	0.0	0.0	4.00	162.38
24.00	2.00	7.00	6.00	5.00	6.00	0.0	0.0	3.00	160.02
24.00	2.00	7.00	6.00	5.00	6.00	0.0	0.0	2.00	99999.94
24.00	2.00	7.00	6.00	5.00	6.00	0.0	0.0	4.00	160.02
24.00	2.00	7.00	6.00	5.00	6.00	0.0	0.0	4.00	160.02
21.50	2.00	7.00	6.00	5.00	6.00	0.0	0.0	4.00	99999.94
24.00	2.00	7.00	6.00	5.00	6.00	0.0	0.0	4.00	160.02

EXAMPLE PROBLEM

EAVE	SLOPE	BASE	PEAK	RAFT	COLU	XBRE	DBRE	A/H	COST
ORTHOGONAL SEARCH									
22.50	2.00	7.00	6.00	5.00	6.00	0.0	0.0	4.00	99999.94
23.50	2.00	7.00	6.00	5.00	6.00	0.0	0.0	4.00	165.47
24.00	2.00	7.00	6.00	5.00	6.00	0.0	0.0	4.00	160.02
24.00	2.00	11.00	6.00	5.00	6.00	0.0	0.0	4.00	167.85
24.00	2.00	9.00	6.00	5.00	6.00	0.0	0.0	4.00	163.37
24.00	2.00	7.00	6.00	5.00	6.00	0.0	0.0	4.00	160.02
24.00	2.00	7.00	6.00	7.50	6.00	0.0	0.0	4.00	168.66
24.00	2.00	7.00	6.00	6.00	6.00	0.0	0.0	4.00	163.48
24.00	2.00	7.00	6.00	5.00	6.00	0.0	0.0	4.00	160.02
24.00	2.00	7.00	6.00	5.00	7.50	0.0	0.0	4.00	163.03
24.00	2.00	7.00	6.00	5.00	6.00	0.0	0.0	4.00	160.02
24.00	2.00	7.00	6.00	5.00	5.00	0.0	0.0	4.00	162.38
24.00	2.00	7.00	6.00	5.00	6.00	0.0	0.0	3.00	160.02
24.00	2.00	7.00	6.00	5.00	6.00	0.0	0.0	2.00	99999.94
24.00	2.00	7.00	6.00	5.00	6.00	0.0	0.0	4.00	160.02
DIAGONAL SEARCH									
23.50	2.00	9.00	6.00	6.00	7.50	0.0	0.0	3.00	166.57
24.00	2.00	9.00	6.00	6.00	7.50	0.0	0.0	3.00	165.74
23.50	2.00	7.00	6.00	6.00	7.50	0.0	0.0	3.00	165.75
24.00	2.00	7.00	6.00	6.00	7.50	0.0	0.0	3.00	166.49
23.50	2.00	9.00	6.00	5.00	7.50	0.0	0.0	3.00	166.95
24.00	2.00	9.00	6.00	5.00	7.50	0.0	0.0	3.00	166.37
23.50	2.00	7.00	6.00	5.00	7.50	0.0	0.0	3.00	165.47
24.00	2.00	7.00	6.00	5.00	7.50	0.0	0.0	3.00	163.03
23.50	2.00	9.00	6.00	6.00	5.00	0.0	0.0	3.00	167.23
24.00	2.00	9.00	6.00	6.00	5.00	0.0	0.0	3.00	165.46
23.50	2.00	7.00	6.00	6.00	5.00	0.0	0.0	3.00	165.75
24.00	2.00	7.00	6.00	6.00	5.00	0.0	0.0	3.00	163.98
23.50	2.00	9.00	6.00	5.00	5.00	0.0	0.0	3.00	166.95
24.00	2.00	9.00	6.00	5.00	5.00	0.0	0.0	3.00	163.87
23.50	2.00	7.00	6.00	5.00	5.00	0.0	0.0	3.00	165.47
24.00	2.00	7.00	6.00	5.00	5.00	0.0	0.0	3.00	160.52
23.50	2.00	9.00	6.00	6.00	6.00	0.0	0.0	3.00	167.23
24.00	2.00	9.00	6.00	6.00	6.00	0.0	0.0	3.00	164.95
23.50	2.00	7.00	6.00	6.00	6.00	0.0	0.0	3.00	165.75
24.00	2.00	7.00	6.00	6.00	6.00	0.0	0.0	3.00	163.48
23.50	2.00	9.00	6.00	5.00	6.00	0.0	0.0	3.00	166.95
24.00	2.00	9.00	6.00	5.00	6.00	0.0	0.0	3.00	163.37
23.50	2.00	7.00	6.00	5.00	6.00	0.0	0.0	3.00	165.47

EXAMPLE PROBLEM

EAVE	SLOPE	BASE	PEAK	RAFT	COLU	XBRE	DBRE	A/H	COST
DIAGONAL SEARCH									
23.50	2.00	9.00	6.00	6.00	7.50	0.0	0.0	4.00	166.57
24.00	2.00	9.00	6.00	6.00	7.50	0.0	0.0	4.00	165.74
23.50	2.00	7.00	6.00	6.00	7.50	0.0	0.0	4.00	165.75
24.00	2.00	7.00	6.00	6.00	7.50	0.0	0.0	4.00	166.49
23.50	2.00	9.00	6.00	5.00	7.50	0.0	0.0	4.00	166.95
24.00	2.00	9.00	6.00	5.00	7.50	0.0	0.0	4.00	166.37
23.50	2.00	7.00	6.00	5.00	7.50	0.0	0.0	4.00	165.47
23.50	2.00	9.00	6.00	6.00	5.00	0.0	0.0	4.00	167.23
24.00	2.00	9.00	6.00	6.00	5.00	0.0	0.0	4.00	165.46
23.50	2.00	7.00	6.00	6.00	5.00	0.0	0.0	4.00	165.75
24.00	2.00	7.00	6.00	6.00	5.00	0.0	0.0	4.00	163.98
23.50	2.00	9.00	6.00	5.00	5.00	0.0	0.0	4.00	166.95
24.00	2.00	9.00	6.00	5.00	5.00	0.0	0.0	4.00	163.87
23.50	2.00	7.00	6.00	5.00	5.00	0.0	0.0	4.00	165.47
23.50	2.00	9.00	6.00	6.00	6.00	0.0	0.0	4.00	166.57
24.00	2.00	9.00	6.00	6.00	6.00	0.0	0.0	4.00	162.73
23.50	2.00	7.00	6.00	6.00	6.00	0.0	0.0	4.00	165.75
23.50	2.00	9.00	6.00	5.00	6.00	0.0	0.0	4.00	166.95

SUMMARY

SPAN	HEIGHT	SLOPE	BAY	DEAD LOAD	LIVE LOAD	WIND LOAD
60.0	14.0	2.0	20.0	4.0	20.0	25.0

WIND LOAD COEFFICIENTS

****WINDWARD****			****LEEWARD*****		
HORIZONTAL		VERTICAL		HORIZONTAL	
WALL	ROOF	ROOF	ROOF	ROOF	WALL
0.70	-0.55	-0.55	-0.50	-0.50	-0.40

CONCENTRATED LOADS

NO	X	Y	HORIZONTAL	VERTICAL	MOMENT	LOCATION
1	10.00	14.38	0.0	5.0	0.0	RIGHT
2	10.00	14.38	0.0	5.0	0.0	LEFT

LOADING COMBINATION NO	1	2
DEAD LOAD	1.00	0.75
LIVE LOAD	1.00	0.0
WIND LOAD	0.0	0.75
CONCENTRATED		
LOAD NO 1	1.00	0.75
LOAD NO 2	1.00	0.75

COLUMN WIDTH	RAFTER WIDTH	A/H RATIO
6.0	5.0	4.0

PLATE THICKNESSES

PART	OUTER FLANGE	WEB	INNER FLANGE	LENGTH	YIELD
1	0.3750	0.1875	0.5000	11.80	42.0
2	0.4375	0.1875	0.6250	6.10	42.0
3	0.1875	0.1875	0.3750	10.99	36.0
4	0.2500	0.1250	0.1875	10.91	36.0

SECTION PROPERTIES

PART	SEC	X	Y	DEPTH	AREA	MOMENT INERTIA	SECTION MODULUS OUTER	SECTION MODULUS INNER
1	1	0.0	0.0	7.00	6.40	59.3	15.41	18.81
1	2	0.12	1.96	9.83	6.93	125.5	23.29	28.21
1	3	0.24	3.93	12.67	7.46	219.3	31.74	38.07
1	4	0.35	5.89	15.50	7.99	342.9	40.75	48.39
1	5	0.47	7.85	18.33	8.52	498.4	50.29	59.18
1	6	0.59	9.81	21.17	9.05	688.1	60.38	70.44
1	7	0.71	11.78	24.00	9.59	914.1	70.99	82.18
2	1	1.54	12.76	24.00	9.61	908.1	69.44	83.13
2	2	2.54	12.95	23.35	9.49	852.7	66.97	80.34
2	3	3.54	13.15	22.69	9.37	799.4	64.53	77.58
2	4	4.54	13.34	22.04	9.25	748.0	62.12	74.84
2	5	5.54	13.53	21.38	9.12	698.7	59.73	72.13
2	6	6.54	13.72	20.73	9.00	651.3	57.38	69.44
2	7	7.54	13.91	20.08	8.88	605.8	55.05	66.78
3	1	7.54	13.91	20.08	6.47	378.8	33.23	43.66
3	2	9.34	14.26	18.90	6.25	328.3	30.48	40.41
3	3	11.14	14.60	17.72	6.03	282.1	27.81	37.24
3	4	12.93	14.95	16.54	5.81	240.1	25.24	34.15
3	5	14.73	15.29	15.37	5.59	202.1	22.76	31.15
3	6	16.53	15.64	14.19	5.37	167.9	20.38	28.23
3	7	18.33	15.98	13.01	5.15	137.5	18.09	25.39
4	1	18.33	15.98	13.01	3.76	109.2	18.21	15.56
4	2	20.12	16.33	11.84	3.61	88.5	16.26	13.82
4	3	21.90	16.67	10.67	3.47	70.2	14.37	12.13
4	4	23.69	17.01	9.51	3.32	54.3	12.53	10.51
4	5	25.47	17.35	8.34	3.17	40.7	10.74	8.95
4	6	27.26	17.70	7.17	3.03	29.2	9.00	7.45
4	7	29.04	18.04	6.00	2.88	19.8	7.32	6.01

DEAD LOAD - FORCES

		LEFT SIDE			RIGHT SIDE		
PART	SEC	AXIAL	SHEAR	MOMENT	AXIAL	SHEAR	MOMENT
1	1	-2.40	-1.91	0.0	-2.40	-1.91	0.0
1	2	-2.40	-1.91	-3.46	-2.40	-1.91	-3.46
1	3	-2.40	-1.91	-6.91	-2.40	-1.91	-6.91
1	4	-2.40	-1.91	-10.37	-2.40	-1.91	-10.37
1	5	-2.40	-1.91	-13.83	-2.40	-1.91	-13.83
1	6	-2.40	-1.91	-17.29	-2.40	-1.91	-17.29
1	7	-2.40	-1.91	-20.76	-2.40	-1.91	-20.76
2	1	-2.25	1.93	-20.71	-2.25	1.93	-20.71
2	2	-2.24	1.85	-18.84	-2.24	1.85	-18.84
2	3	-2.23	1.77	-17.05	-2.23	1.77	-17.05
2	4	-2.21	1.70	-15.34	-2.21	1.70	-15.34
2	5	-2.20	1.62	-13.71	-2.20	1.62	-13.71
2	6	-2.19	1.54	-12.16	-2.19	1.54	-12.16
2	7	-2.17	1.46	-10.69	-2.17	1.46	-10.69
3	1	-2.17	1.46	-10.69	-2.17	1.46	-10.69
3	2	-2.15	1.32	-8.24	-2.15	1.32	-8.24
3	3	-2.13	1.18	-6.06	-2.13	1.18	-6.06
3	4	-2.10	1.03	-4.13	-2.10	1.03	-4.13
3	5	-2.08	0.89	-2.46	-2.08	0.89	-2.46
3	6	-2.06	0.75	-1.05	-2.06	0.75	-1.05
3	7	-2.03	0.61	0.10	-2.03	0.61	0.10
4	1	-2.03	0.61	0.10	-2.03	0.61	0.10
4	2	-2.01	0.47	0.99	-2.01	0.47	0.99
4	3	-1.99	0.33	1.62	-1.99	0.33	1.62
4	4	-1.96	0.18	2.00	-1.96	0.18	2.00
4	5	-1.94	0.04	2.12	-1.94	0.04	2.12
4	6	-1.92	-0.10	1.98	-1.92	-0.10	1.98
4	7	-1.89	-0.24	1.60	-1.89	-0.24	1.60

REACTIONS	LEFT	RIGHT
VERTICAL	2.40	2.40
HORIZONTAL	-1.91	-1.91

LIVE LOAD - FORCES

		LEFT SIDE			RIGHT SIDE		
PART	SEC	AXIAL	SHEAR	MOMENT	AXIAL	SHEAR	MOMENT
1	1	-12.00	-9.53	0.0	-12.00	-9.53	0.0
1	2	-12.00	-9.53	-17.28	-12.00	-9.53	-17.28
1	3	-12.00	-9.53	-34.57	-12.00	-9.53	-34.57
1	4	-12.00	-9.53	-51.87	-12.00	-9.53	-51.87
1	5	-12.00	-9.53	-69.17	-12.00	-9.53	-69.17
1	6	-12.00	-9.53	-86.47	-12.00	-9.53	-86.47
1	7	-12.00	-9.53	-103.78	-12.00	-9.53	-103.78
2	1	-11.27	9.66	-103.53	-11.27	9.66	-103.53
2	2	-11.20	9.27	-94.18	-11.20	9.27	-94.18
2	3	-11.14	8.87	-85.24	-11.14	8.87	-85.24
2	4	-11.07	8.48	-76.69	-11.07	8.48	-76.69
2	5	-11.01	8.08	-68.54	-11.01	8.08	-68.54
2	6	-10.94	7.69	-60.79	-10.94	7.69	-60.79
2	7	-10.87	7.30	-53.44	-10.87	7.30	-53.44
3	1	-10.87	7.30	-53.44	-10.87	7.30	-53.44
3	2	-10.76	6.59	-41.21	-10.76	6.59	-41.21
3	3	-10.64	5.88	-30.28	-10.64	5.88	-30.28
3	4	-10.52	5.17	-20.64	-10.52	5.17	-20.64
3	5	-10.40	4.46	-12.30	-10.40	4.46	-12.30
3	6	-10.28	3.75	-5.25	-10.28	3.75	-5.25
3	7	-10.16	3.04	0.51	-10.16	3.04	0.51
4	1	-10.16	3.04	0.51	-10.16	3.04	0.51
4	2	-10.05	2.33	4.94	-10.05	2.33	4.94
4	3	-9.93	1.63	8.10	-9.93	1.63	8.10
4	4	-9.81	0.92	9.98	-9.81	0.92	9.98
4	5	-9.69	0.22	10.59	-9.69	0.22	10.59
4	6	-9.58	-0.48	9.92	-9.58	-0.48	9.92
4	7	-9.46	-1.19	7.98	-9.46	-1.19	7.98

REACTIONS	LEFT	RIGHT
VERTICAL	12.00	12.00
HORIZONTAL	-9.53	-9.53

EXAMPLE PROBLEM PAGE 5

WIND LOAD - FORCES

		LEFT SIDE			RIGHT SIDE		
PART	SEC	AXIAL	SHEAR	MOMENT	AXIAL	SHEAR	MOMENT
1	1	8.97	10.73	0.0	6.80	3.16	0.0
1	2	8.97	10.04	19.33	6.80	3.55	5.79
1	3	8.97	9.36	37.32	6.80	3.94	12.35
1	4	8.97	8.67	53.96	6.80	4.34	19.68
1	5	8.97	7.98	69.25	6.80	4.73	27.79
1	6	8.97	7.30	83.21	6.80	5.12	36.67
1	7	8.97	6.61	95.81	6.80	5.52	46.32
2	1	6.82	-7.53	94.98	6.63	-5.40	46.07
2	2	6.83	-7.25	87.76	6.63	-5.15	40.98
2	3	6.83	-6.96	80.81	6.64	-4.89	36.13
2	4	6.84	-6.68	74.12	6.65	-4.64	31.52
2	5	6.85	-6.40	67.69	6.65	-4.38	27.15
2	6	6.85	-6.12	61.53	6.66	-4.13	23.02
2	7	6.86	-5.84	55.63	6.67	-3.88	19.13
3	1	6.86	-5.84	55.63	6.67	-3.88	19.13
3	2	6.87	-5.34	45.69	6.68	-3.42	12.74
3	3	6.89	-4.83	36.60	6.69	-2.96	7.12
3	4	6.90	-4.33	28.38	6.70	-2.50	2.29
3	5	6.91	-3.82	21.01	6.71	-2.04	-1.77
3	6	6.92	-3.32	14.50	6.72	-1.59	-5.05
3	7	6.93	-2.81	8.85	6.73	-1.13	-7.55
4	1	6.93	-2.81	8.85	6.73	-1.13	-7.55
4	2	6.95	-2.31	4.10	6.74	-0.67	-9.25
4	3	6.96	-1.81	0.19	6.75	-0.22	-10.20
4	4	6.97	-1.31	-2.88	6.77	0.23	-10.37
4	5	6.98	-0.81	-5.09	6.78	0.69	-9.78
4	6	7.00	-0.31	-6.46	6.79	1.14	-8.42
4	7	7.01	0.19	-6.98	6.80	1.60	-6.29

REACTIONS

	LEFT	RIGHT
VERTICAL	-8.97	-6.80
HORIZONTAL	10.73	3.16

CONCENTRATED LOAD NO. 1 - FORCES

		LEFT SIDE			RIGHT SIDE		
PART	SEC	AXIAL	SHEAR	MOMENT	AXIAL	SHEAR	MOMENT
1	1	-0.86	-1.43	0.0	-4.14	-1.43	0.0
1	2	-0.86	-1.43	-2.71	-4.14	-1.43	-2.32
1	3	-0.86	-1.43	-5.42	-4.14	-1.43	-4.64
1	4	-0.86	-1.43	-8.12	-4.14	-1.43	-6.96
1	5	-0.86	-1.43	-10.83	-4.14	-1.43	-9.28
1	6	-0.86	-1.43	-13.54	-4.14	-1.43	-11.60
1	7	-0.86	-1.43	-16.25	-4.14	-1.43	-13.92
2	1	-1.55	0.61	-16.94	-2.09	3.85	-11.88
2	2	-1.55	0.61	-16.36	-2.09	3.85	-8.01
2	3	-1.55	0.61	-15.77	-2.09	3.85	-4.15
2	4	-1.55	0.61	-15.19	-2.09	3.85	-0.29
2	5	-1.55	0.61	-14.60	-2.09	3.85	3.57
2	6	-1.55	0.61	-14.02	-2.09	3.85	7.44
2	7	-1.55	0.61	-13.43	-2.09	3.85	11.30
3	1	-1.55	0.61	-13.43	-2.09	3.85	11.30
3	2	-1.55	0.61	-12.38	-2.09	3.85	18.25
3	3	-1.55	0.61	-11.33	-1.27	-1.08	19.52
3	4	-1.55	0.61	-10.28	-1.27	-1.08	17.48
3	5	-1.55	0.61	-9.23	-1.27	-1.08	15.44
3	6	-1.55	0.61	-8.17	-1.27	-1.08	13.40
3	7	-1.55	0.61	-7.12	-1.27	-1.08	11.36
4	1	-1.55	0.61	-7.12	-1.27	-1.08	11.36
4	2	-1.55	0.61	-6.08	-1.27	-1.08	9.34
4	3	-1.55	0.61	-5.03	-1.27	-1.08	7.31
4	4	-1.55	0.61	-3.99	-1.27	-1.08	5.29
4	5	-1.55	0.61	-2.94	-1.27	-1.08	3.27
4	6	-1.55	0.61	-1.90	-1.27	-1.08	1.24
4	7	-1.55	0.61	-0.86	-1.27	-1.08	-0.78

REACTIONS

	LEFT	RIGHT
VERTICAL	0.86	4.14
HORIZONTAL	-1.43	-1.43

CONCENTRATED LOAD NO. 2 - FORCES

		LEFT SIDE			RIGHT SIDE		
PART	SEC	AXIAL	SHEAR	MOMENT	AXIAL	SHEAR	MOMENT
1	1	-4.14	-1.43	0.0	-0.86	-1.43	0.0
1	2	-4.14	-1.43	-2.32	-0.86	-1.43	-2.71
1	3	-4.14	-1.43	-4.64	-0.86	-1.43	-5.42
1	4	-4.14	-1.43	-6.96	-0.86	-1.43	-8.12
1	5	-4.14	-1.43	-9.28	-0.86	-1.43	-10.83
1	6	-4.14	-1.43	-11.60	-0.86	-1.43	-13.54
1	7	-4.14	-1.43	-13.92	-0.86	-1.43	-16.25
2	1	-2.09	3.85	-11.88	-1.55	0.61	-16.94
2	2	-2.09	3.85	-8.01	-1.55	0.61	-16.36
2	3	-2.09	3.85	-4.15	-1.55	0.61	-15.77
2	4	-2.09	3.85	-0.29	-1.55	0.61	-15.19
2	5	-2.09	3.85	3.57	-1.55	0.61	-14.60
2	6	-2.09	3.85	7.44	-1.55	0.61	-14.02
2	7	-2.09	3.85	11.30	-1.55	0.61	-13.43
3	1	-2.09	3.85	11.30	-1.55	0.61	-13.43
3	2	-2.09	3.85	18.25	-1.55	0.61	-12.38
3	3	-1.27	-1.08	19.52	-1.55	0.61	-11.33
3	4	-1.27	-1.08	17.48	-1.55	0.61	-10.28
3	5	-1.27	-1.08	15.44	-1.55	0.61	-9.23
3	6	-1.27	-1.08	13.40	-1.55	0.61	-8.17
3	7	-1.27	-1.08	11.36	-1.55	0.61	-7.12
4	1	-1.27	-1.08	11.36	-1.55	0.61	-7.12
4	2	-1.27	-1.08	9.34	-1.55	0.61	-6.08
4	3	-1.27	-1.08	7.31	-1.55	0.61	-5.03
4	4	-1.27	-1.08	5.29	-1.55	0.61	-3.99
4	5	-1.27	-1.08	3.27	-1.55	0.61	-2.94
4	6	-1.27	-1.08	1.24	-1.55	0.61	-1.90
4	7	-1.27	-1.08	-0.78	-1.55	0.61	-0.86

REACTIONS

	LEFT	RIGHT
VERTICAL	4.14	0.86
HORIZONTAL	-1.43	-1.43

COMBINATION NO. 1 - FORCES

		LEFT SIDE			RIGHT SIDE		
PART	SEC	AXIAL	SHEAR	MOMENT	AXIAL	SHEAR	MOMENT
1	1	-19.40	-14.29	0.0	-19.40	-14.29	0.0
1	2	-19.40	-14.29	-25.77	-19.40	-14.29	-25.77
1	3	-19.40	-14.29	-51.54	-19.40	-14.29	-51.54
1	4	-19.40	-14.29	-77.33	-19.40	-14.29	-77.33
1	5	-19.40	-14.29	-103.12	-19.40	-14.29	-103.12
1	6	-19.40	-14.29	-128.91	-19.40	-14.29	-128.91
1	7	-19.40	-14.29	-154.71	-19.40	-14.29	-154.71
2	1	-17.17	16.06	-153.05	-17.17	16.06	-153.05
2	2	-17.09	15.58	-137.39	-17.09	15.58	-137.39
2	3	-17.01	15.11	-122.21	-17.01	15.11	-122.21
2	4	-16.93	14.64	-107.50	-16.93	14.64	-107.50
2	5	-16.85	14.16	-93.28	-16.85	14.16	-93.28
2	6	-16.77	13.69	-79.53	-16.77	13.69	-79.53
2	7	-16.69	13.22	-66.26	-16.69	13.22	-66.26
3	1	-16.69	13.22	-66.26	-16.69	13.22	-66.26
3	2	-16.55	12.37	-43.59	-16.55	12.37	-43.59
3	3	-15.59	6.58	-28.14	-15.59	6.58	-28.14
3	4	-15.45	5.73	-17.56	-15.45	5.73	-17.56
3	5	-15.30	4.88	-8.54	-15.30	4.88	-8.54
3	6	-15.16	4.03	-1.07	-15.16	4.03	-1.07
3	7	-15.02	3.18	4.85	-15.02	3.18	4.85
4	1	-15.02	3.18	4.85	-15.02	3.18	4.85
4	2	-14.88	2.33	9.19	-14.88	2.33	9.19
4	3	-14.74	1.48	12.00	-14.74	1.48	12.00
4	4	-14.60	0.64	13.28	-14.60	0.64	13.28
4	5	-14.46	-0.21	13.03	-14.46	-0.21	13.03
4	6	-14.32	-1.05	11.25	-14.32	-1.05	11.25
4	7	-14.18	-1.90	7.94	-14.18	-1.90	7.94

REACTIONS

	LEFT	RIGHT
VERTICAL	19.40	19.40
HORIZONTAL	-14.29	-14.29

COMBINATION NO. 1 - STRESSES

		LEFT SIDE				RIGHT SIDE			
				BENDING				BENDING	
PART	SEC	AXIAL	SHEAR	OUTER	INNER	AXIAL	SHEAR	OUTER	INNER
1	1	-3.0	-12.4	0.0	0.0	-3.0	-12.4	0.0	0.0
1	2	-2.8	-8.3	13.3	-11.0	-2.8	-8.3	13.3	-11.0
1	3	-2.6	-6.2	19.5	-16.4	-2.6	-6.2	19.5	-16.4
1	4	-2.4	-5.0	22.8	-19.3	-2.4	-5.0	22.8	-19.3
1	5	-2.3	-4.2	24.6	-21.1	-2.3	-4.2	24.6	-21.1
1	6	-2.1	-3.6	25.6	-22.1	-2.1	-3.6	25.6	-22.1
1	7	-2.0	-3.1	26.2	-22.8	-2.0	-3.1	26.2	-22.8
2	1	-1.8	3.7	26.4	-22.1	-1.8	3.7	26.4	-22.1
2	2	-1.8	3.7	24.6	-20.6	-1.8	3.7	24.6	-20.6
2	3	-1.8	3.7	22.7	-18.9	-1.8	3.7	22.7	-18.9
2	4	-1.8	3.7	20.8	-17.3	-1.8	3.7	20.8	-17.3
2	5	-1.8	3.7	18.7	-15.5	-1.8	3.7	18.7	-15.5
2	6	-1.9	3.7	16.6	-13.8	-1.9	3.7	16.6	-13.8
2	7	-1.9	3.7	14.4	-11.9	-1.9	3.7	14.4	-11.9
3	1	-2.6	3.6	23.9	-18.2	-2.6	3.6	23.9	-18.2
3	2	-2.6	3.6	17.2	-13.0	-2.6	3.6	17.2	-13.0
3	3	-2.6	2.0	12.1	-9.1	-2.6	2.0	12.1	-9.1
3	4	-2.7	1.9	8.4	-6.2	-2.7	1.9	8.4	-6.2
3	5	-2.7	1.7	4.5	-3.3	-2.7	1.7	4.5	-3.3
3	6	-2.8	1.6	0.6	-0.5	-2.8	1.6	0.6	-0.5
3	7	-2.9	1.4	-3.2	2.3	-2.9	1.4	-3.2	2.3
4	1	-4.0	2.0	-3.2	3.7	-4.0	2.0	-3.2	3.7
4	2	-4.1	1.7	-6.8	8.0	-4.1	1.7	-6.8	8.0
4	3	-4.3	1.2	-10.0	11.9	-4.3	1.2	-10.0	11.9
4	4	-4.4	0.6	-12.7	15.2	-4.4	0.6	-12.7	15.2
4	5	-4.6	-0.1	-14.6	17.5	-4.6	-0.1	-14.6	17.5
4	6	-4.7	-1.2	-15.0	18.1	-4.7	-1.2	-15.0	18.1
4	7	-4.9	-2.6	-13.0	15.9	-4.9	-2.6	-13.0	15.9

EXAMPLE PROBLEM

COMBINATION NO. 2 - FORCES

		LEFT SIDE			RIGHT SIDE		
PART	SEC	AXIAL	SHEAR	MOMENT	AXIAL	SHEAR	MOMENT
1	1	1.18	4.47	0.0	-0.45	-1.21	0.0
1	2	1.18	3.96	8.13	-0.45	-0.91	-2.02
1	3	1.18	3.44	15.26	-0.45	-0.62	-3.47
1	4	1.18	2.93	21.37	-0.45	-0.32	-4.34
1	5	1.18	2.41	26.48	-0.45	-0.03	-4.62
1	6	1.18	1.90	30.58	-0.45	0.27	-4.33
1	7	1.18	1.38	33.66	-0.45	0.56	-3.45
2	1	0.69	-0.85	34.10	0.55	0.74	-2.59
2	2	0.71	-0.70	33.42	0.56	0.88	-1.67
2	3	0.72	-0.55	32.88	0.58	1.01	-0.63
2	4	0.74	-0.40	32.48	0.59	1.14	0.53
2	5	0.75	-0.24	32.21	0.60	1.27	1.81
2	6	0.77	-0.09	32.09	0.62	1.40	3.21
2	7	0.78	0.06	32.11	0.63	1.53	4.73
3	1	0.78	0.06	32.11	0.63	1.53	4.73
3	2	0.81	0.33	32.49	0.66	1.77	7.77
3	3	1.45	-3.10	29.05	1.30	-1.69	6.95
3	4	1.48	-2.82	23.59	1.33	-1.45	4.02
3	5	1.50	-2.55	18.58	1.35	-1.22	1.49
3	6	1.53	-2.28	14.01	1.38	-0.98	-0.65
3	7	1.56	-2.01	9.90	1.41	-0.74	-2.40
4	1	1.56	-2.01	9.90	1.41	-0.74	-2.40
4	2	1.59	-1.74	6.26	1.43	-0.51	-3.75
4	3	1.61	-1.47	3.07	1.46	-0.27	-4.72
4	4	1.64	-1.20	0.32	1.48	-0.04	-5.31
4	5	1.67	-0.93	-1.99	1.51	0.20	-5.50
4	6	1.69	-0.66	-3.85	1.54	0.43	-5.32
4	7	1.72	-0.39	-5.27	1.56	0.67	-4.75

REACTIONS

	LEFT	RIGHT
VERTICAL	-1.18	0.45
HORIZONTAL	4.47	-1.21

COMBINATION NO. 2 - STRESSES

		LEFT SIDE				RIGHT SIDE			
				BENDING				BENDING	
PART	SEC	AXIAL	SHEAR	OUTER	INNER	AXIAL	SHEAR	OUTER	INNER
1	1	0.2	3.9	0.0	0.0	-0.1	-1.1	0.0	0.0
1	2	0.2	2.3	-4.2	3.5	-0.1	-0.5	1.0	-0.9
1	3	0.2	1.5	-5.8	4.8	-0.1	-0.3	1.3	-1.1
1	4	0.1	1.0	-6.3	5.3	-0.1	-0.1	1.3	-1.1
1	5	0.1	0.7	-6.3	5.4	-0.1	-0.0	1.1	-0.9
1	6	0.1	0.5	-6.1	5.2	-0.0	0.1	0.9	-0.7
1	7	0.1	0.3	-5.7	5.0	-0.0	0.1	0.6	-0.5
2	1	0.1	-0.2	-5.9	4.9	0.1	0.2	0.4	-0.4
2	2	0.1	-0.2	-6.0	5.0	0.1	0.2	0.3	-0.3
2	3	0.1	-0.1	-6.1	5.1	0.1	0.2	0.1	-0.1
2	4	0.1	-0.1	-6.3	5.2	0.1	0.3	-0.1	0.1
2	5	0.1	-0.0	-6.5	5.4	0.1	0.3	-0.4	0.3
2	6	0.1	-0.0	-6.7	5.6	0.1	0.4	-0.7	0.6
2	7	0.1	0.0	-7.0	5.8	0.1	0.4	-1.0	0.9
3	1	0.1	0.0	-11.6	8.8	0.1	0.4	-1.7	1.3
3	2	0.1	0.1	-12.8	9.7	0.1	0.5	-3.1	2.3
3	3	0.2	-0.9	-12.5	9.4	0.2	-0.5	-3.0	2.2
3	4	0.3	-0.9	-11.2	8.3	0.2	-0.5	-1.9	1.4
3	5	0.3	-0.9	-9.8	7.2	0.2	-0.4	-0.8	0.6
3	6	0.3	-0.9	-8.2	6.0	0.3	-0.4	0.4	-0.3
3	7	0.3	-0.8	-6.6	4.7	0.3	-0.3	1.6	-1.1
4	1	0.4	-1.3	-6.5	7.6	0.4	-0.5	1.6	-1.9
4	2	0.4	-1.2	-4.6	5.4	0.4	-0.4	2.8	-3.3
4	3	0.5	-1.1	-2.6	3.0	0.4	-0.2	3.9	-4.7
4	4	0.5	-1.1	-0.3	0.4	0.4	-0.1	5.1	-6.1
4	5	0.5	-0.9	2.2	-2.7	0.5	0.2	6.2	-7.4
4	6	0.6	-0.8	5.1	-6.2	0.5	0.5	7.1	-8.6
4	7	0.6	-0.6	8.6	-10.5	0.5	0.9	7.8	-9.5

ALLOWABLE STRESSES

PART	SEC	SHEAR	COMP.	TENSION	BENDING OUTER	BENDING INNER
1	1	16.80	25.20	25.20	25.20	25.20
1	2	16.80	23.76	25.20	25.20	25.20
1	3	15.63	21.98	25.20	25.20	25.20
1	4	12.61	21.83	25.20	25.20	25.20
1	5	9.59	21.68	25.20	25.20	25.20
1	6	7.10	22.99	25.20	25.20	25.20
1	7	5.47	24.26	25.20	25.20	25.20
2	1	5.56	23.30	25.20	23.69	25.20
2	2	5.89	22.26	25.20	23.70	25.20
2	3	6.25	22.29	25.20	23.71	25.20
2	4	6.64	21.35	25.20	23.72	25.20
2	5	7.08	21.39	25.20	23.82	25.20
2	6	7.56	21.42	25.20	24.58	25.20
2	7	8.09	21.46	25.20	25.20	25.20
3	1	7.68	17.93	21.60	20.21	21.60
3	2	8.69	17.10	21.60	19.71	20.09
3	3	9.93	17.22	21.60	19.76	20.87
3	4	10.68	17.34	21.60	19.80	21.60
3	5	11.53	17.45	21.60	19.85	21.60
3	6	12.53	17.57	21.60	19.90	21.60
3	7	13.71	14.23	21.60	19.94	16.01
4	1	8.22	14.91	21.60	20.19	15.60
4	2	9.98	15.12	21.60	20.82	15.73
4	3	11.12	15.33	21.60	21.60	15.85
4	4	12.55	15.54	21.60	21.60	15.97
4	5	14.40	15.76	21.60	21.60	16.10
4	6	14.40	15.98	21.60	21.60	16.22
4	7	14.40	20.76	21.60	21.60	21.60

```
PURLINS ARE LOCATED AT X=
  28.18   23.18   18.18   13.18    8.18    3.94
  -0.29

GIRTS ARE LOCATED AT Y=
   0.50    3.00    8.00   11.00   14.00


INNER FLANGE BRACES REQUIRED

ON RAFTER AT X=
  28.18   18.18   13.18    8.18    3.94   -0.29

ON COLUMN AT Y=
   3.00    8.00   14.00


WEB STIFFENERS REQUIRED

ON RAFTER AT X=
NONE REQUIRED

ON COLUMN AT Y=
NONE REQUIRED
```

```
C       MAIN CONTROL PROGRAM
        COMMON OPTL(6),A(9,17),OPSAV(10)
        COMMON PLTM(10),FLAN(10),GIRT(10),WC(6)
        COMMON WEBPL(6),WEB(10),NOCOL,NORAF,NOTOT
        COMMON KUTE,NOFLA,NOWEB,NPS,BAY,GD,NOGRT,H,SPAN,DL,VL
        COMMON WL,WFVP,FIRST,SPACE,NOCBM,KPLT,KWIND
        COMMON CBM(4,7),FLTI(10),FLTO(10),PLCOR(10)
        COMMON FY1,FY2,COST1,COST2,NOALT,KALT(10)
        COMMON TABLE(4,7),BRACE(5),ITLE(76),LANA,LDES,LPAT
        COMMON SCOS1,SCOS2,IMOT
        DIMENSION X(10,7),Y(10,7),D(10,7),NHOLD(10),PARTL(10)
        DIMENSION ERT(10,7),AREA(10,7),SO(10,7),SI(10,7)
        DIMENSION SHL(10,7),SHR(10,7),AXL(10,7),AXR(10,7)
        DIMENSION SML(10,7),SMR(10,7),R(4),C(6)
        DIMENSION SHEAR(8,10,7),OMENT(8,10,7),AXIAL(8,10,7),
       □REACT(4,4)
        DIMENSION PURL(20),MAST(20)
        DIMENSION UBLI(10,7),UBLO(10,7)
        DIMENSION WIDO(10),WIDI(10)
        DIMENSION III(4)
        DIMENSION KAST(20),TEMP(20),TEMG(10)
        DIMENSION STIFF(2,10),Q(10,7)
        DIMENSION SOLTA(100,10)
        EQUIVALENCE (BD,A(3,2)),(COLU,A(6,2)),(ED,A(1,2))
        EQUIVALENCE (RD,A(4,2)),(RAFT,A(5,2)),(SLOPE,A(2,2))
        EQUIVALENCE(XBRE,A(7,2)),(DBRE,A(8,2)),(AOH,A(9,2))
        NOPT=0
        SWTCH=0.0
        NOSOL=0
        NEXT=1
        SCOS1=0.0
        KUTE=0
        ICTL=40
        C(1)=0.0
        C(2)=0.0
        C(3)=1.0
        C(4)=1.0
        C(5)=0.0
        C(6)=0.0
        IPAGE=0
        CALL NITIAL
        ALPHA=ATAN(SLOPE/12.0)
        CALL LOCATE(FIRST,SPACE,PURL,NOPUR,SPAN,GD,SLOPE,
       □BRACE,MAST)
        KIN=4
        KIT=4
        IF(KUTE)2,1,2
      1 KIN=1
        KIT=0
        CALL OPTMUM(KIN)
        IF(KIN-4)306,307,306
```

```
307 LPAT=1
306 CONTINUE
  2 CALL GEOM(H,SPAN,BD,ED,RD,SLOPE,GD,X,Y,D,PLTM,NPS,XE,
   ¤YE,XR, YR,NOCOL,NORAF,NOTOT,PARTL,KPLT,PLCOR,XBRE,
   ¤DBRE,NORFA)
    WTSA2=0.0
    WTSAV=0.0
    NORAF=NOTOT-NOCOL
    CALL UNSUL(X,Y,NOGRT,GIRT,NOPUR,PURL,NOCOL,NOTOT,UBLI,
   ¤UBLO, H,GD,SLOPE,D,BD,SPAN,FIRST)
    DO 11 I=1,NOTOT
    FY=FY1
    IF(NOALT)201,200,201
201 DO 203 J=1,NOALT
    IF(I-KALT(J))203,204,203
203 CONTINUE
    GO TO 200
204 FY=FY2
200 CONTINUE
    DO 11 J=1,7
 11 CALL INERT(D(I,J),FLTO(I),WEB(I),FLTI(I),WIDO(I),
   ¤WIDI(I), AREA(I,J),SO(I,J),SI(I,J),ERT(I,J),NOCOL,FY,
   ¤I)
 80 DO 31 I=1,4
    DO 31 J=1,4
 31 REACT(I,J)=0.0
    IF(KIT-4)300,301,300
301 IF(LDES-1)300,600,300
600 IPAGE=IPAGE+1
    WRITE(3,303)(ITLE(I),I=1,51),IPAGE
303 FORMAT(I1,50A1,'    PAGE',I3)
    WRITE(3,302)SPAN,H,SLOPE,BAY,DL,VL,WL,(WC(I),I=1,6)
302 FORMAT(/T28,'SUMMARY'//T36,'DEAD     LIVE     WIND'/ ' '
   ¤,'  SPAN    HEIGHT    SLOPE     BAY     LOAD     LOAD    '
   ¤,' LOAD'/ F7.1,F9.1,F8.1,F7.1,3F8.1//T20,'WIND LOAD'
   ¤,' COEFFICIENTS'/ T14,'****WINDWARD****   ****LEEWAR'
   ¤,'D*****'/ T14,'HORIZONTAL    VERTICAL    HORIZONTAL'/
   ¤T14,'WALL   ROOF   ROOF   ROOF   ROOF  WALL'/T12,6F6.2)
    KTRIG=0
    DO 320 L=1,4
    XC=TABLE(L,1)
    YC=TABLE(L,2)
    FV=TABLE(L,3)
    FH=TABLE(L,4)
    FM=TABLE(L,5)
    ID=TABLE(L,6)
    IF(ABS(FV)+ABS(FH)+ABS(FM))321,320,321
321 IF(KTRIG)322,323,322
323 WRITE(3,324)
324 FORMAT(/T22,'CONCENTRATED LOADS'/ '   NO      X       Y   '
   ¤,' HORIZONTAL   VERTICAL     MOMENT      LOCATION')
```

```
      KTRIG=KTRIG+1
  322 GO TO (331,332,331,332),ID
  331 DO 333 I=1,NOCOL
      IF(YC-Y(I,7))334,333,333
  333 CONTINUE
      STOP 333
  334 YI=YC
      XI=X(I,1)+(YI-Y(I,1))/(Y(I,7)-Y(I,1))*(X(I,7)-X(I,1))
      FMI=FM
      IF(XC)335,336,335
  335 FMI=FMI+FV*(XC-XI)
      GO TO 336
  332 M=NOCOL+1
      DO 337 I=M,NOTOT
      IF(XC-X(I,7))338,337,337
  337 CONTINUE
      STOP 337
  338 XI=XC
      YI=Y(I,1)+(XI-X(I,1))/(X(I,7)-X(I,1))*(Y(I,7)-Y(I,1))
      FMI=FM
      IF(YC)339,336,339
  339 FMI=FMI-FH*(YI-YC)
  336 GO TO (341,341,342,342),ID
  341 WRITE(3,343)L,XI,YI,FH,FV,FMI
  343 FORMAT(I4,F7.2,F6.2,-3PF11.1,F10.1,F10.1,'         LEFT'
     □)
      GO TO 320
  342 WRITE(3,344)L,XI,YI,FH,FV,FMI
  344 FORMAT(I4,F7.2,F6.2,-3PF11.1,F10.1,F10.1,'        RIGHT'
     □)
  320 CONTINUE
      DO 346 I=1,4
  346 III(I)=I
      WRITE(3,345)(III(I),I=1,NOCBM)
  345 FORMAT(/' LOADING'/' COMBINATION NO',I7,3I8)
      WRITE(3,347)(CBM(I,1),I=1,NOCBM)
  347 FORMAT(' DEAD LOAD',5X,4F8.2)
      WRITE(3,348)(CBM(I,2),I=1,NOCBM)
  348 FORMAT(' LIVE LOAD',5X,4F8.2)
      WRITE(3,349)(CBM(I,3),I=1,NOCBM)
  349 FORMAT(' WIND LOAD',5X,4F8.2)
      IF(KTRIG)351,352,351
  351 WRITE(3,350)
  350 FORMAT(' CONCENTRATED')
      DO 353 I=1,4
      L=I+3
      SUM=0.0
      DO 354 J=1,4
  354 SUM=SUM+ABS(CBM(J,L))
      IF(SUM)355,353,355
  355 WRITE(3,356)I,(CBM(M,L),M=1,NOCBM)
```

```
356 FORMAT(' LOAD NO',I2,5X,4F8.2)
353 CONTINUE
352 CONTINUE
    WRITE(3,360)COLU,RAFT,AOH
360 FORMAT(/' COLUMN  RAFTER  A/H'/' WIDTH   WIDTH  '
   ¤,' RATIO'/ F5.1,2F8.1)
    WRITE(3,362)
362 FORMAT(/'           PLATE THICKNESSES'/'        OUTER '
   ¤,'            INNER'/' PART   FLANGE   WEB    FLANGE  '
   ¤,' LENGTH    YIELD')
    DO 365 I=1,NOTOT
    FY=FY1
    IF(NOALT)211,210,211
211 DO 213 J=1,NOALT
    IF(I-KALT(J))213,214,213
213 CONTINUE
    GO TO 210
214 FY=FY2
210 CONTINUE
365 WRITE(3,366)I,FLTO(I),WEB(I),FLTI(I),PARTL(I),FY
366 FORMAT(I4,F10.4,2F8.4,F9.2,-3PF9.1)
    IPAGE=IPAGE+1
    WRITE(3,303)(ITLE(I),I=1,51),IPAGE
    WRITE(3,310)
310 FORMAT(/T23,'SECTION PROPERTIES'/ T39,'MOMENT '
   ¤,' SECTION MODULUS'/ ' PART SEC    X      Y    DEPTH '
   ¤,'  AREA INERTIA   OUTER   INNER')
    DO 312 I=1,NOTOT
    DO 311 J=1,7
311 WRITE(3,313)I,J,X(I,J),Y(I,J),D(I,J),AREA(I,J),ERT(I,
   ¤J),SO(I,J), SI(I,J)
313 FORMAT(' ',2I4,4F7.2,F8.1,2F8.2)
    WRITE(3,315)
315 FORMAT(' ')
    IF(I-5)312,314,312
314 IF(NOTOT-5)666,312,666
666 IPAGE=IPAGE+1
    WRITE(3,303)(ITLE(K),K=1,51),IPAGE
    WRITE(3,310)
312 CONTINUE
300 CONTINUE
    DO 30 I=1,8
    DO 30 J=1,10
    DO 30 K=1,7
    SHEAR(I,J,K)=0.0
    AXIAL(I,J,K)=0.0
 30 OMENT(I,J,K)=0.0
    IF(DL) 32,33,32
 32 W=DL*BAY
    CALL UNIF (SPAN,H,SLOPE,BD,GD,X,Y,W,C,PARTL,NOCOL,
   ¤NORAF, NOTOT,ERT,AXL,AXR,SHL,SHR,SML,SMR,R,ED)
```

```
      IF(KIT-4)126,127,126
  127 IF(LANA-1)126,128,126
  128 CALL PRINT(IPAGE,ITLE,1,AXL,SHL,SML,AXR,SHR,SMR,NOTOT,
     ¤L,R)
  126 CONTINUE
      CALL COMBIN(AXL,AXR,SHL,SHR,SML,SMR,R,CBM,SHEAR,AXIAL,
     ¤OMENT,REACT,1,NOTOT,NOCBM)
   33 IF(VL)34,35,34
   34 IF(DL)37,36,37
   37 FACT=VL/DL
      DO 38 I=1,NOTOT
      DO 38 J=1,7
      SHL(I,J)=SHL(I,J)*FACT
      SHR(I,J)=SHR(I,J)*FACT
      AXL(I,J)=AXL(I,J)*FACT
      AXR(I,J)=AXR(I,J)*FACT
      SML(I,J)=SML(I,J)*FACT
   38 SMR(I,J)=SMR(I,J)*FACT
      DO 39 I=1,4
   39 R(I)=R(I)*FACT
      GO TO 40
   36 W=VL*BAY
      CALL UNIF (SPAN,H,SLOPE,BD,GD,X,Y,W,C,PARTL,NOCOL,
     ¤NORAF, NOTOT,ERT,AXL,AXR,SHL,SHR,SML,SMR,R,ED)
   40 CONTINUE
      IF(KIT-4)129,130,129
  130 IF(LANA-1)129,131,129
  131 CALL PRINT(IPAGE,ITLE,2,AXL,SHL,SML,AXR,SHR,SMR,NOTOT,
     ¤L,R)
  129 CONTINUE
      CALL COMBIN(AXL,AXR,SHL,SHR,SML,SMR,R,CBM,SHEAR,AXIAL,
     ¤OMENT,REACT,2,NOTOT,NOCBM)
   35 IF(WL)41,42,41
   41 W=WL*BAY
      CALL UNIF (SPAN,H,SLOPE,BD,GD,X,Y,W,WC,PARTL,NOCOL,
     ¤NORAF, NOTOT,ERT,AXL,AXR,SHL,SHR,SML,SMR,R,ED)
      IF(KIT-4)132,133,132
  133 IF(LANA-1)132,134,132
  134 CALL PRINT(IPAGE,ITLE,3,AXL,SHL,SML,AXR,SHR,SMR,NOTOT,
     ¤L,R)
  132 CONTINUE
      CALL COMBIN(AXL,AXR,SHL,SHR,SML,SMR,R,CBM,SHEAR,AXIAL,
     ¤OMENT,REACT,3,NOTOT,NOCBM)
   42 DO 50 L=1,4
      XC=TABLE(L,1)
      YC=TABLE(L,2)
      FV=TABLE(L,3)
      FH=TABLE(L,4)
      FM=TABLE(L,5)
      ID=TABLE(L,6)
      IF(ABS(FV)+ABS(FH)+ABS(FM))43,50,43
```

```
 43 CALL CONCEN (SPAN,H,SLOPE,BD,GD,X,Y,XC,YC,FV,FH,FM,ID,
   ¤PARTL,NOCOL,NORAF,NOTOT,ERT,AXL,AXR,SHL,SHR,SML,SMR,R,
   ¤ED)
    IF(KIT-4)135,136,135
136 IF(LANA-1)135,137,135
137 CALL PRINT(IPAGE,ITLE,4,AXL,SHL,SML,AXR,SHR,SMR,NOTOT,
   ¤L,R)
135 CONTINUE
    LL=L+3
    CALL COMBIN(AXL,AXR,SHL,SHR,SML,SMR,R,CBM,SHEAR,AXIAL,
   ¤OMENT,REACT,LL,NOTOT,NOCBM)
 50 CONTINUE
    IF(KIT-4)140,141,140
141 IF(LANA-1)140,142,140
142 DO 144 L=1,NOCBM
    IRT=NOCBM+L
    DO 143 I=1,NOTOT
    DO 143 J=1,7
    AXL(I,J)=AXIAL(L,I,J)
    SHL(I,J)=SHEAR(L,I,J)
    SML(I,J)=OMENT(L,I,J)
    AXR(I,J)=AXIAL(IRT,I,J)
    SHR(I,J)=SHEAR(IRT,I,J)
143 SMR(I,J)=OMENT(IRT,I,J)
    DO 444 I=1,4
444 R(I)=REACT(L,I)
    CALL PRINT(IPAGE,ITLE,5,AXL,SHL,SML,AXR,SHR,SMR,NOTOT,
   ¤L,R)
144 CALL OUTPUT(IPAGE,ITLE,AXL,SHL,SML,AXR,SHR,SMR,NOTOT,
   ¤L, D,SO,SI,AREA,FLTO,FLTI,WEB,WIDI,WIDO,PARTL)
140 CONTINUE
    IF(KIT-4)95,91,95
 95 KASTER=0
    DO 60 I=1,NOTOT
    FY=FY1
    IF(NOALT)221,220,221
221 DO 223 J=1,NOALT
    IF(I-KALT(J))223,224,223
223 CONTINUE
    GO TO 220
224 FY=FY2
220 CONTINUE
 60 CALL SELECT(NOFLA,NOWEB,FLAN,WEBPL,WEB,NOCBM,FLTI,
   ¤FLTO,D,ERT,AREA,SO,SI,SHEAR,OMENT,AXIAL,UBLO,UBLI,AOH,
   ¤FY,I, PARTL,WIDI,WIDO,KASTER,NOCOL)
    IF(KASTER)76,75,76
 76 WEGHT=99999.99
    OPFUN=99999.99
    GO TO 973
 75 WEGHT=0.0
    OPFUN=0.0
```

```
      DO 68 I=1,NOTOT
      UNIT=COST1
      IF(NOALT)914,913,914
  914 DO 911 M=1,NOALT
      IF(KALT(M)-I)911,912,911
  911 CONTINUE
      GO TO 913
  912 UNIT=COST2
  913 CONTINUE
      WID=COLU
      IF(I-NOCOL)968,968,69
   69 WID=RAFT
  968 PRWGH=((FLTI(I)+FLTO(I))*WID+(D(I,1)+D(I,7))/2.0
     ¤*WEB(I))*6.8 *PARTL(I)
      WEGHT=WEGHT+PRWGH
   68 OPFUN=OPFUN+UNIT*PRWGH
   83 IF(WEGHT-WTSAV)72,93,72
   72 IF(WEGHT-WTSA2)3400,3401,3400
 3400 WTSA2=WTSAV
      WTSAV=WEGHT
      GO TO 80
 3401 IF(WTSA2-WTSAV)3400,3402,3402
 3402 WTSAV=WTSA2
      GO TO 93
   93 IF(KIN-4)73,94,73
   94 KIT=4
      GO TO 80
   73 CONTINUE
 4735 IF(AOH-3.0)4750,4750,4751
 4751 NOSTR=0
      NOSTC=0
      GO TO 4790
 4750 KK=NOCBM*2
      DO 4754 I=1,NOTOT
      TAN=(D(I,1)-D(I,7))/PARTL(I)/12.0
      DO 4754 J=1,7
      SHL(I,J)=0.0
      DO 4754 K=1,KK
      VT=OMENT(K,I,J)/SI(I,J)*FLTI(I)*WIDI(I)*TAN
      VCOR=ABS(SHEAR(K,I,J)+VT)
      IF(SHL(I,J)-VCOR)4756,4756,4754
 4756 SHL(I,J)=VCOR
 4754 CONTINUE
      CALL STFNER (AOH,FLTI,FLTO,D,WEB,FY1,FY2,SHL,Q,NOSTR,
     ¤STIFF, NOTOT,NOCOL,X,Y,ALPHA,NOALT,KALT,NOSTC,XBRE)
 4790 NOSTF=NOSTR+NOSTC
      OPFUN=OPFUN+SCOS1*NOSTF
  973 OPTL(3)=OPFUN
   66 FORMAT(' ',9F6.2,F9.2)
  285 IF(LPAT-1)81,82,81
   82 IF(ICTL-40)84,85,84
```

```
 85 ICTL=0
    WRITE(3,87)(ITLE(M),M=1,51)
    IF(SWTCH)603,602,603
603 WRITE(3,293)
    GO TO 604
602 WRITE(3,298)
604 ICTL=ICTL+1
 87 FORMAT(I1,50A1/'0  EAVE SLOPE  BASE   PEAK   RAFT   COLU'
   □,'  XBRE  DBRE   A/H  COST')
 84 ICTL=ICTL+1
    WRITE(3,66)(A(I,2),I=1,9),OPFUN
 81 IF(KIN-4)90,91,90
 90 IF(NOPT)275,287,275
287 DO 250 I=1,9
250 SOLTA(NEXT,I)=A(I,2)
    SOLTA(NEXT,10)=OPTL(3)
    NEXT=NEXT+1
    IF(NOSOL-NEXT)253,254,254
253 NOSOL=NEXT-1
254 IF(NEXT-101)275,251,275
251 NEXT=1
275 NOPT=0
    IF(SWTCH)256,255,256
255 CALL OPTMUM(KIN)
    GO TO 257
256 CALL SEARCH(SWTCH)
    IF(SWTCH)258,259,270
258 KIN=4
    DO 281 I=1,9
281 A(I,2)=OPSAV(I)
    GO TO 67
259 KIN=2
    IF(LPAT-1) 296,297,296
297 WRITE(3,298)
    ICTL=ICTL+1
298 FORMAT(' ORTHOGONAL SEARCH')
296 CONTINUE
    DO 280 I=1,9
280 A(I,2)=OPSAV(I)
    NOV=OPTL(1)
    DO 261 I=1,NOV
    IF(A(I,1))262,261,262
261 CONTINUE
262 OPTL(2)=OPSAV(I)
    OPTL(3)=OPSAV(10)
    OPTL(4)=I
    GO TO 255
257 IF(KIN-4) 270,260,270
260 KIN=5
    NOV=OPTL(1)
    SUM=0.0
```

```
      DO 668 I=1,NOV
      IF(A(I,1))669,668,669
  669 SUM=SUM+1.0
  668 CONTINUE
      IF(SUM-1.0)258,258,670
  670 IF(IMOT)258,671,258
  671 IF(LPAT-1)290,291,290
  291 WRITE(3,293)
      ICTL=ICTL+1
  293 FORMAT(' DIAGONAL SEARCH')
  290 CONTINUE
      GO TO 256
  270 NOV=OPTL(1)
      DO 276 I=1,NOSOL
      DO 272 J=1,NOV
      IF(A(J,2)-SOLTA(I,J))276,272,276
  272 CONTINUE
      GO TO 273
  276 CONTINUE
      GO TO 2
  273 OPTL(3)=SOLTA(I,10)
      OPFUN=OPTL(3)
      NOPT=1
      GO TO 285
   67 DO 92 I=1,9
   92 A(I,2)=OPSAV(I)
      GO TO 2
   91 DO 700 I=1,NOPUR
  700 KAST(I)=-I
      MAST(NOPUR)=-NOPUR
      NT=1
  703 IF(MAST(NT))701,702,702
  701 IF(NT-NOPUR+1)704,705,705
  704 NT=NT+1
      GO TO 703
  702 KAST(NT)=NT
      NOTEM=0
      DO 708 I=1,NOPUR
      IF(KAST(I))707,708,708
  707 NOTEM=NOTEM+1
      TEMP(NOTEM)=PURL(I)
  708 CONTINUE
      DO 710 I=1,NOPUR
      IF(KAST(I))711,710,710
  710 CONTINUE
  711 TFIRS=PURL(I)*12.0/COS(ALPHA)
      CALL UNSUL(X,Y,NOGRT,GIRT,NOTEM,TEMP,NOCOL,NOTOT,UBLI,
     □AXL, H,GD,SLOPE,D,BD,SPAN,TFIRS)
      KK=NOCOL+1
      DO 720 I=KK,NOTOT
      KASTR=0
```

```
      FY=FY1
      DO 714 M=1,NOALT
      IF(I-KALT(M))714,713,714
  713 FY=FY2
      GO TO 715
  714 CONTINUE
  715 CALL CHECK(WEB,NOCBM,FLTI,FLTO,D,ERT,AREA,SO,SI,SHEAR,
     ¤OMENT, AXIAL,UBLO,UBLI,AOH,FY,I,PARTL,WIDI,WIDO,KASTR)
      IF(KASTR)721,720,721
  720 CONTINUE
      GO TO 704
  721 KAST(NT)=-NT
      MAST(NT)=-NT
      GO TO 704
  705 NOTEM=0
      DO 718 I=1,NOPUR
      IF(KAST(I))717,718,718
  717 NOTEM=NOTEM+1
      TEMP(NOTEM)=PURL(I)
  718 CONTINUE
      DO 725 I=1,NOPUR
      IF(MAST(I))726,725,725
  725 CONTINUE
  726 TFIRS=PURL(I)*12.0/COS(ALPHA)
      DO 722 I=1,NOGRT
  722 KAST(I)=-I
      NT=0
  731 NT=NT+1
      IF(NT-NOGRT+1)734,734,735
  734 KAST(NT)=NT
      NOTMG=0
      DO 738 I=1,NOGRT
      IF(KAST(I))737,738,738
  737 NOTMG=NOTMG+1
      TEMG(NOTMG)=GIRT(I)
  738 CONTINUE
      CALL UNSUL(X,Y,NOTMG,TEMG,NOTEM,TEMP,NOCOL,NOTOT,UBLI,
     ¤AXL, H,GD,SLOPE,D,BD,SPAN,TFIRS)
      DO 740 I=1,NOCOL
      KASTR=0
      FY=FY1
      DO 744 M=1,NOALT
      IF(I-KALT(M))744,743,744
  743 FY=FY2
      GO TO 745
  744 CONTINUE
  745 CALL CHECK(WEB,NOCBM,FLTI,FLTO,D,ERT,AREA,SO,SI,SHEAR,
     ¤OMENT, AXIAL,UBLO,UBLI,AOH,FY,I,PARTL,WIDI,WIDO,KASTR)
      IF(KASTR)741,740,741
  740 CONTINUE
      GO TO 731
```

```
741 KAST(NT)=-NT
    GO TO 731
735 IF(AOH-3.0)750,750,751
751 DO 752 I=1,NOTOT
    DO 752 J=1,7
752 Q(I,J)=AOH
    NOSTR=0
    NOSTC=0
    GO TO 790
750 KK=NOCBM*2
    DO 754 I=1,NOTOT
    TAN=(D(I,1)-D(I,7))/PARTL(I)/12.0
    DO 754 J=1,7
    SHL(I,J)=0.0
    DO 754 K=1,KK
    VT=OMENT(K,I,J)/SI(I,J)*FLTI(I)*WIDI(I)*TAN
    VCOR=ABS(SHEAR(K,I,J)+VT)
    IF(SHL(I,J)-VCOR)756,756,754
756 SHL(I,J)=VCOR
754 CONTINUE
    CALL STFNER (AOH,FLTI,FLTO,D,WEB,FY1,FY2,SHL,Q,NOSTR,
   ¤STIFF, NOTOT,NOCOL,X,Y,ALPHA,NOALT,KALT,NOSTC,XBRE)
    IF(NOSTC)757,758,757
758 DO 759 I=1,NOCOL
    DO 759 J=1,7
759 Q(I,J)=99.9
    GO TO 761
757 STIFF(1,NOSTC+1)=0.0
    DO 762 I=1,NOCOL
    DO 762 J=1,7
    DO 763 M=1,NOSTC
    L=NOSTC-M+1
    IF(Y(I,J)-STIFF(1,L))764,765,763
764 Q(I,J)=(STIFF(1,L)-STIFF(1,L+1))*12.0/(D(I,J)-FLTI(I)
   ¤-FLTO(I))
    GO TO 762
765 IF(L-1)767,764,767
767 L=L-1
    GO TO 764
763 CONTINUE
    STOP 763
762 CONTINUE
761 KK=NOCOL+1
    IF(NOSTR)770,771,770
771 DO 772 I=KK,NOTOT
    DO 772 J=1,7
772 Q(I,J)=99.9
    GO TO 790
770 STIFF(2,NOSTR+1)=X(NOTOT,7)
    N=NOSTR+1
    DO 773 I=KK,NOTOT
```

```
      DO 773 J=1,7
      DO 774 M=1,N
      IF(X(I,J)-STIFF(2,M))775,775,774
  775 IF(M-1)773,774,773
  774 CONTINUE
  773 Q(I,J)=(STIFF(2,M)-STIFF(2,M-1))*12.0/COS(ALPHA)/(D(I,
     ¤J)-FLTO(I) -FLTI(I))
  790 IF(LANA-1)812,820,812
  820 ICTL=5
      DO 810 I=1,NOTOT
      IF(ICTL-5)811,822,811
  822 ICTL=0
      IPAGE=IPAGE+1
      WRITE(3,303)(ITLE(M),M=1,51),IPAGE
      WRITE(3,791)
  791 FORMAT('OALLOWABLE STRESSES'/'0',T40,'BENDING'/ ''
     ¤,' PART SEC    SHEAR    COMP. TENSION    OUTER    INNER')
  811 ICTL=ICTL+1
      DO 809 J=1,7
      FY=FY1
      DO 813 K=1,NOALT
      IF(I-KALT(K))813,814,813
  813 CONTINUE
      GO TO 815
  814 FY=FY2
  815 CALL STRESS(UBLO(I,J),UBLI(I,J),D(I,J),FLTO(I),FLTI(I)
     ¤,WIDI(I), WIDO(I),WEB(I),FBO,FBI,FA,FT,FY)
      CALL SSTRES (Q(I,J),D(I,J),FLTO(I),FLTI(I),WEB(I),FY,
     ¤FV)
  809 WRITE(3,807)I,J,FV,FA,FT,FBO,FBI
  807 FORMAT(2I4,-3P5F8.2)
  810 WRITE(3,315)
  812 IF(LDES-1)450,601,450
  601 IPAGE=IPAGE+1
      WRITE(3,303)(ITLE(M),M=1,51),IPAGE
      DIST=SPAN/2.0-(GD+BD/2.0)/12.0
      DO 415 I=1,NOPUR
  415 PURL(I)=DIST-PURL(I)
      DO 416 I=1,NOTEM
  416 TEMP(I)=DIST-TEMP(I)
      WRITE(3,401)(PURL(M),M=1,NOPUR)
  401 FORMAT('OPURLINS ARE LOCATED AT X='/(6F8.2))
      WRITE(3,402)(GIRT(M),M=1,NOGRT)
  402 FORMAT('OGIRTS ARE LOCATED AT Y='/(6F8.2))
      WRITE(3,403)(TEMP(M),M=1,NOTEM)
  403 FORMAT(/'OINNER FLANGE BRACES REQUIRED'/'OON RAFTER'
     ¤,' AT X='/ (6F8.2))
      WRITE(3,404)(TEMG(M),M=1,NOTMG)
  404 FORMAT('OON COLUMN AT Y='/(6F8.2))
      WRITE(3,409)
  409 FORMAT(/'OWEB STIFFENERS REQUIRED'/'OON RAFTER AT X=')
```

```
      IF(NOSTR)420,421,420
  421 WRITE(3,410)
  410 FORMAT(' NONE REQUIRED')
      GO TO 425
  420 WRITE(3,406)(STIFF(2,M),M=1,NOSTR)
  406 FORMAT(6F8.2)
  425 WRITE(3,405)
  405 FORMAT('0ON COLUMN AT Y=')
      IF(NOSTC)430,431,430
  431 WRITE(3,410)
      GO TO 450
  430 WRITE(3,406)(STIFF(1,M),M=1,NOSTC)
  450 STOP 450
      END
```

```
      SUBROUTINE BSTRES(CC,RUL,DLOA,FY,FLT,H,WEB,WID,FB)
      IF(RUL-40.0)2,3,3
    2 FB=0.6*FY
      GO TO 5
    3 FB=(1.0-RUL**2/(2.0*CC**2))*0.6*FY
      FBC=12000000.0/DLOA
      IF(FBC-FB)4,4,6
    6 FB=FBC
    4 IF(FB-0.6*FY)5,5,7
    7 FB=FY*0.6
    5 IF(H/WEB-24000.0/SQRT(FB))8,8,9
    9 FB=FB*(1.0-0.0005*H*WEB/(FLT*WID)*(H/WEB-24000.0
     □/SQRT(FB)))
    8 RETURN
      END
```

```
      SUBROUTINE CHECK(WEB,NOCBM,FLTI,FLTO,D,ERT,AREA,SO,SI,
     □SHEAR, OMENT,AXIAL,UBLO,UBLI,AOH,FY,I,PARTL,WIDI,WIDO,
     □KASTR)
      DIMENSION WEB(10),FLTI(10),FLTO(10),D(10,7),ERT(10,7),
     □AREA(10,7), SO(10,7),SI(10,7),SHEAR(8,10,7),OMENT(8,
     □10,7),AXIAL(8,10,7), UBLO(10,7),UBLI(10,7),PARTL(10),
     □WIDI(10),WIDO(10)
      M=2*NOCBM
      TAN=(D(I,1)-D(I,7))/12.0/PARTL(I)
      SECT=1.0/COS(ATAN(TAN))
      DO 13 J=1,7
      CALL STRESS(UBLO(I,J),UBLI(I,J),D(I,J),FLTO(I),FLTI(I)
     □,WIDI(I), WIDO(I),WEB(I),FBO,FBI,FA,FT,FY)
      DO 13 K=1,M
      SBO=OMENT(K,I,J)/SO(I,J)*(-12.0)
      SA=AXIAL(K,I,J)/AREA(I,J)
      IF(SBO)15,16,16
   15 DUM=SBO/FBO
      GO TO 17
   16 DUM=SBO/FT
   17 IF(SA)18,19,19
   18 DUM=DUM+SA/FA
      GO TO 20
   19 DUM=DUM+SA/FT
   20 IF(ABS(DUM)-1.0)60,60,23
   60 SBI=OMENT(K,I,J)/SI(I,J)*12.0*SECT
      SA=AXIAL(K,I,J)/AREA(I,J)
      IF(SBI)45,46,46
   45 DUM=SBI/FBI
      GO TO 47
   46 DUM=SBI/FT
   47 IF(SA)48,49,49
   48 DUM=DUM+SA/FA
      GO TO 50
   49 DUM=DUM+SA/FT
   50 IF(ABS(DUM)-1.0)13,13,23
   13 CONTINUE
      RETURN
   23 KASTR=9999
      RETURN
      END
```

```
      SUBROUTINE COMBIN(AXL,AXR,SHL,SHR,SML,SMR,R,CBM,SHEAR,
     ¤AXIAL,OMENT,REACT,N,NOTOT,NOCBM)
      DIMENSION SHL(10,7),SHR(10,7),AXL(10,7),AXR(10,7)
      DIMENSION SHEAR(8,10,7),OMENT(8,10,7),AXIAL(8,10,7),
     ¤REACT(4,4)
      DIMENSION SML(10,7),SMR(10,7),R(4),CBM(4,7)
      DO 5 K=1,NOCBM
      M=K+NOCBM
      FACT=CBM(K,N)
      DO 4 I=1,NOTOT
      DO 4 J=1,7
      SHEAR(K,I,J)=SHEAR(K,I,J)+SHL(I,J)*FACT
      AXIAL(K,I,J)=AXIAL(K,I,J)+AXL(I,J)*FACT
      OMENT(K,I,J)=OMENT(K,I,J)+SML(I,J)*FACT
      SHEAR(M,I,J)=SHEAR(M,I,J)+SHR(I,J)*FACT
      AXIAL(M,I,J)=AXIAL(M,I,J)+AXR(I,J)*FACT
    4 OMENT(M,I,J)=OMENT(M,I,J)+SMR(I,J)*FACT
      DO 5 I=1,4
    5 REACT(K,I)=REACT(K,I)+R(I)*FACT
      RETURN
      END
```

```
      SUBROUTINE CONCEN (SPAN,H,SLOPE,BD,GD,X,Y,XC,YC,FV,FH,
     ¤FM,ID,PLGH, NOCOL,NORAF,NOTOT,ERT,AXL,AXR,SHL,SHR,SML,
     ¤SMR,R,ED)
      DIMENSION X(10,7),Y(10,7),PLGH(10),ERT(10,7),AXL(10,7)
     ¤, AXR(10,7),SHL(10,7),SHR(10,7),SML(10,7),SMR(10,7),
     ¤R(4)
      DIMENSION C(7)
      ALPHA=ATAN(SLOPE/12.0)
      SINA=SIN(ALPHA)
      COSA=COS(ALPHA)
      C(1)=1.0
      C(2)=4.0
      C(3)=2.0
      C(4)=4.0
      C(5)=2.0
      C(6)=4.0
      C(7)=1.0
      GO TO (1,2,1,2),ID
    1 DO 3 I=1,NOCOL
      IF(YC-Y(I,7))4,3,3
    3 CONTINUE
      STOP 3
    4 YI=YC
      XI=X(I,1)+(YI-Y(I,1))/(Y(I,7)-Y(I,1))*(X(I,7)-X(I,1))
      FMI=FM
      IF(XC)5,6,5
    5 FMI=FMI+FV*(XC-XI)
      GO TO 6
    2 K=NOCOL+1
      DO 7 I=K,NOTOT
      IF(XC-X(I,7))8,7,7
    7 CONTINUE
      STOP 7
    8 XI=XC
      YI=Y(I,1)+(XI-X(I,1))/(X(I,7)-X(I,1))*(Y(I,7)-Y(I,1))
      FMI=FM
      IF(YC)9,6,9
    9 FMI=FMI-FH*(YI-YC)
    6 DIST=SPAN-(2.0*GD+BD)/12.0
      R(2)=(FH*YI+FV*XI+FMI)/DIST
      R(1)=FV-R(2)
      R(3)=FH
      DO 10 I=1,NOCOL
      DO 10 J=1,7
      SML(I,J)=R(3)*Y(I,J)+R(1)*X(I,J)
      SMR(I,J)=R(2)*X(I,J)
      AXL(I,J)=-R(1)
      AXR(I,J)=-R(2)
      SHL(I,J)=R(3)
      IF(YI-Y(I,J))11,10,10
   11 SML(I,J)=SML(I,J)+FMI-FH*(Y(I,J)-YI)-FV*(X(I,J)-XI)
```

```
      SHL(I,J)=SHL(I,J)-FH
      AXL(I,J)=AXL(I,J)+FV
   10 CONTINUE
      K=NOCOL+1
      DO 12 I=K,NOTOT
      DO 12 J=1,7
      SML(I,J)=R(3)*Y(I,J)+R(1)*X(I,J)
      SMR(I,J)=R(2)*X(I,J)
      AXL(I,J)=-R(1)
      AXR(I,J)=-R(2)
      SHL(I,J)=R(3)
      IF(XI-X(I,J))20,12,12
   20 SML(I,J)=SML(I,J)+FMI-FH*(Y(I,J)-YI)-FV*(X(I,J)-XI)
      SHL(I,J)=SHL(I,J)-FH
      AXL(I,J)=AXL(I,J)+FV
   12 CONTINUE
      SUMM=0.0
      SUMY=0.0
      DO 13 I=1,NOTOT
      DO 13 J=1,7
      YDS=Y(I,J)*PLGH(I)/6.0/ERT(I,J)
      SUMM=SUMM+(SML(I,J)+SMR(I,J))*YDS*C(J)
   13 SUMY=SUMY+2.0*YDS*Y(I,J)*C(J)
      DS=ED*SLOPE/24.0
      SUMM=SUMM/3.0+(SML(NOCOL,7)+SMR(NOCOL,7))*(Y(NOCOL,7)
     ¤+DS/2.0) *DS/ERT(NOCOL,7)
      SUMY=SUMY/3.0+2.0*(Y(NOCOL,7)+DS/2.0)**2*DS/ERT(NOCOL,
     ¤7)
      SUMM=SUMM+(SML(NOCOL+1,1)+SMR(NOCOL+1,1))*(Y(NOCOL+1,
     ¤1)- DS/2.0*SINA)*DS/ERT(NOCOL+1,1)
      SUMY=SUMY+2.0*(Y(NOCOL+1,1)-DS/2.0*SINA)**2*DS
     ¤/ERT(NOCOL+1,1)
      R(4)=-SUMM/SUMY
      R(3)=R(3)+R(4)
      DO 14 I=1,NOTOT
      DO 14 J=1,7
      SML(I,J)=SML(I,J)+Y(I,J)*R(4)
      SMR(I,J)=SMR(I,J)+Y(I,J)*R(4)
      SHL(I,J)=SHL(I,J)+R(4)
   14 SHR(I,J)=R(4)
      DO 15 I=K,NOTOT
      DO 15 J=1,7
      HORL=SHL(I,J)
      HORR=SHR(I,J)
      VRTL=AXL(I,J)
      VRTR=AXR(I,J)
      AXL(I,J)=HORL*COSA+VRTL*SINA
      AXR(I,J)=HORR*COSA+VRTR*SINA
      SHR(I,J)=HORR*SINA-VRTR*COSA
   15 SHL(I,J)=HORL*SINA-VRTL*COSA
      GO TO (16,16,17,17),ID
```

```
17 DO 18 I=1,NOTOT
   DO 18 J=1,7
   Z=SHL(I,J)
   SHL(I,J)=SHR(I,J)
   SHR(I,J)=Z
   Z=SML(I,J)
   SML(I,J)=SMR(I,J)
   SMR(I,J)=Z
   Z=AXL(I,J)
   AXL(I,J)=AXR(I,J)
18 AXR(I,J)=Z
   Z=R(1)
   R(1)=R(2)
   R(2)=Z
   Z=R(3)
   R(3)=R(4)
   R(4)=Z
16 RETURN
   END
```

```
      SUBROUTINE CSTRES(CC,RUL,FY,FA)
      IF(RUL-CC)1,1,2
    1 FS=5.0/3.0+3.0/8.0*RUL/CC-RUL**3/(8.0*CC**3)
      FA=(1.0-RUL**2/(2.0*CC**2))*FY/FS
      GO TO 3
    2 FA=149000000.0/RUL**2
    3 RETURN
      END
```

```
      SUBROUTINE FINDER(WEB,WEBPL,NOWEB,KWEB)
      DIMENSION WEBPL(6)
      DO 1 I=1,NOWEB
      IF(WEB-WEBPL(I))1,2,1
    1 CONTINUE
      STOP 501
    2 KWEB=I
      RETURN
      END
```

```
      SUBROUTINE GEOM(H,SPAN,BD,ED,RD,SLOPE,GD,X,Y,D,PLTM,
     ¤NPS,XE,YE,XR, YR,NOCOL,NORAF,NOTOT,PARTL,KPLT,PLCOR,
     ¤XBRE,DBRE,NORFA)
      DIMENSION PLCOR(10)
      DIMENSION X(10,7),Y(10,7),D(10,7),PLTM(10),NHOLD(10),
     ¤PARTL(10)
      ALPHA=ATAN(SLOPE/12.0)
      BETA=(1.570796-ALPHA)/2.0
      BTAN=SIN(BETA)/COS(BETA)
      SINA=SIN(ALPHA)
      COSA=COS(ALPHA)
      HCLR=H-(GD+ED)*BTAN/12.0
      HCOL=H-GD*BTAN/12.0
      COLGH=SQRT(((ED-BD)/12.0)**2+HCLR**2)
      DUMMY=COLGH+ED*BTAN/12.0
      IF(KPLT)27,26,27
   27 DO 28 I=1,10
      PLTM(I)=0.0
   28 NHOLD(I)=0
      DO 29 I=1,NOCOL
      J=NOCOL-I+1
      NHOLD(J)=1
   29 PLTM(J)=PLCOR(J)
      NPS=NOCOL
      GO TO 30
   26 CALL PARTS(DUMMY,NPS,PLTM,NHOLD)
      NOCOL=0
      DO 1 K=1,NPS
    1 NOCOL=NOCOL+NHOLD(K)
   30 Y(NOCOL,7)=HCLR
      X(NOCOL,7)=(ED-BD)/24.0
      N=1
    4 IF(NHOLD(N))3,2,3
    2 N=N+1
      GO TO 4
    3 Y(NOCOL,1)=HCOL-PLTM(N)
      X(NOCOL,1)=Y(NOCOL,1)/HCLR*X(NOCOL,7)
      NHOLD(N)=NHOLD(N)-1
      IF(NOCOL-1)9,9,5
    9 X(1,1)=0.0
      Y(1,1)=0.0
      GO TO 6
    5 DO 7 I=2,NOCOL
      J=NOCOL-I+1
      Y(J,7)=Y(J+1,1)
      X(J,7)=X(J+1,1)
      IF(NOCOL-I)9,9,8
    8 IF(NHOLD(N))10,11,10
   11 N=N+1
      GO TO 8
   10 NHOLD(N)=NHOLD(N)-1
```

```
      Y(J,1)=Y(J,7)-PLTM(N)*HCLR/COLGH
    7 X(J,1)=X(NOCOL,7)*Y(J,1)/HCLR
    6 YT=H+SPAN/2.0*SINA
      XR=SPAN/2.0-(BD/2.0+GD)/12.0
      XO=X(NOCOL,7)+ED/24.0
      YO=HCLR
      YB=YT-(GD+RD)/(12.0*COSA)
      XE=X(NOCOL,7)
      YE=Y(NOCOL,7) +ED*BTAN/24.0
      YR=YB+RD*COSA/24.0
      DUMMY=YE-Y(NOCOL,7)
      X(NOCOL+1,1)=XE+DUMMY*COSA
      Y(NOCOL+1,1)=YE+DUMMY*SINA
      RAFL=SQRT((XR-XO)**2+(YB-HCLR)**2)
      DUMMY=RAFL+(RD*SINA+ED*BTAN)/12.0
      IF(KPLT)31,32,31
   31 DO 33 I=1,10
      PLTM(I)=0.0
   33 NHOLD(I)=0
      DO 34 I=1,NORAF
      J=NOTOT-I+1
      PLTM(I)=PLCOR(J)
   34 NHOLD(I)=1
      NPS=NORAF
      GO TO 35
   32 IF(XBRE)332,332,232
  232 DUMMX=XBRE+(BD/2.0+GD)/12.0
      DUMMY=H+DUMMX*SINA/COSA
      YBRE=DUMMY-(DBRE/2.0+GD)/12.0/COSA
      XI=XBRE+DBRE/12.0*SINA
      YI=YBRE-DBRE/12.0*COSA
      RAFLA=SQRT((XI-XO)**2+(YI-YO)**2)
      DUMMY=RAFLA+ED*BTAN/12.0
      CALL PARTS (DUMMY,NPS,PLTM,NHOLD)
      NORFA=0
      DO 50 K=1,NPS
   50 NORFA=NORFA+NHOLD(K)
      NOSUB=NOCOL+NORFA
      Y(NOSUB,7)=YBRE
      X(NOSUB,7)=XBRE
      DUMMY=YBRE-Y(NOCOL+1,1)
      DUMMX=XBRE-X(NOCOL+1,1)
      DUMMZ=ATAN(DUMMY/DUMMX)
      COSE=COS(DUMMZ)
      SINE=SIN(DUMMZ)
      RATIO=SQRT(DUMMY**2+DUMMX**2)/RAFLA
      IF(NORFA-1)52,57,52
   52 N=1
   53 IF(NHOLD(N))54,55,54
   55 N=N+1
      GO TO 53
```

```
 54 NHOLD(N)=NHOLD(N)-1
    DUMMY=PLTM(N)*RATIO
    Y(NOSUB,1)=Y(NOSUB,7)-SINE*DUMMY
    X(NOSUB,1)=X(NOSUB,7)-COSE*DUMMY
    DO 58 I=2,NORFA
    J=NOSUB-I+1
    Y(J,7)=Y(J+1,1)
    X(J,7)=X(J+1,1)
    IF(NORFA-I)57,57,60
 60 IF(NHOLD(N))61,62,61
 62 N=N+1
    GO TO 60
 61 NHOLD(N)=NHOLD(N)-1
    DUMMY=PLTM(N)*RATIO
    Y(J,1)=Y(J,7)-SINE*DUMMY
 58 X(J,1)=X(J,7)-SINE*DUMMY
 57 RAFLB=SQRT((XR-XI)**2+(YR-YI)**2)
    DUMMY=RAFLB+RD*SINA/12.0
    X(NOSUB+1,1)=X(NOSUB,7)
    Y(NOSUB+1,1)=Y(NOSUB,7)
    CALL PARTS(DUMMY,NPS,PLTM,NHOLD)
    NORFB=0
    DO 70 K=1,NPS
 70 NORFB=NORFB+NHOLD(K)
    NOTOT=NOSUB+NORFB
    Y(NOTOT,7)=YR
    X(NOTOT,7)=XR
    DUMMY=YR-YBRE
    DUMMX=XR-XBRE
    DUMMZ=ATAN(DUMMY/DUMMX)
    COSE=COS(DUMMZ)
    SINE=SIN(DUMMZ)
    RATIO=SQRT(DUMMY**2+DUMMX**2)/RAFLB
    L=NORFB
    GO TO 72
332 CALL PARTS(DUMMY,NPS,PLTM,NHOLD)
    NORAF=0
    DO 12 K=1,NPS
 12 NORAF=NORAF+NHOLD(K)
    NOTOT=NORAF+NOCOL
 35 Y(NOTOT,7)=YR
    X(NOTOT,7)=XR-RD*SINA/24.0
    DUMMY=Y(NOTOT,7)-Y(NOCOL+1,1)
    DUMMX=X(NOTOT,7)-X(NOCOL+1,1)
    DUMMZ=ATAN(DUMMY/DUMMX)
    COSE=COS(DUMMZ)
    SINE=SIN(DUMMZ)
    RATIO=SQRT(DUMMY**2+DUMMX**2)/RAFL
    L=NORAF
 72 IF(L-1)16,17,16
 16 N=1
```

```
 13 IF(NHOLD(N))14,15,14
 15 N=N+1
    GO TO 13
 14 NHOLD(N)=NHOLD(N)-1
    DUMMY=(PLTM(N)-RD*SINA/12.0)*RATIO
    Y(NOTOT,1)=Y(NOTOT,7)-SINE*DUMMY
    X(NOTOT,1)=X(NOTOT,7)-COSE*DUMMY
    DO 18 I=2,L
    J=NOTOT-I+1
    Y(J,7)=Y(J+1,1)
    X(J,7)=X(J+1,1)
    IF(L-I)17,17,20
 20 IF(NHOLD(N))21,22,21
 22 N=N+1
    GO TO 20
 21 NHOLD(N)=NHOLD(N)-1
    DUMMY=PLTM(N)*RATIO
    Y(J,1)=Y(J,7)-SINE*DUMMY
 18 X(J,1)=X(J,7)-COSE*DUMMY
 17 DO 23 I=1,NOTOT
    PARTL(I)=SQRT((X(I,7)-X(I,1))**2+(Y(I,7)-Y(I,1))**2)
    DO 24 J=1,7,6
    IF(I-NOCOL)25,25,46
 25 D(I,J)=BD+(ED-BD)*Y(I,J)/HCLR
    GO TO 24
 46 IF(XBRE)73,446,73
 73 IF(I-NOCOL-NORFA)75,75,76
 75 D(I,J)=ED-(ED-DBRE)*(X(I,J)-X(NOCOL+1,1))/(XBRE
   ¤-X(NOCOL+1,1))
    GO TO 24
 76 D(I,J)=DBRE-(DBRE-RD)*(X(I,J)-XBRE)/(X(NOTOT,7)-XBRE)
    GO TO 24
446 D(I,J)=ED-(ED-RD)*(X(I,J)-X(NOCOL+1,1))/(X(NOTOT,7)
   ¤-X(NOCOL+1,1))
 24 CONTINUE
    DO 23 J=2,6
    X(I,J)=X(I,1)+(X(I,7)-X(I,1))/6.0*(J-1)
    Y(I,J)=Y(I,1)+(Y(I,7)-Y(I,1))/6.0*(J-1)
 23 D(I,J)=D(I,1)+(D(I,7)-D(I,1))/6.0*(J-1)
    RETURN
    END
```

```
      SUBROUTINE INERT(D,TOUT,WEB,TIN,BOUT,BIN,AREA,SO,SI,
     *ERT,NOCOL,FY,I)
      COMMON OPTL(6),A(9,17)
      EQUIVALENCE (COLU,A(6,2)),(RAFT,A(5,2))
      WID=COLU
      IF(I-NOCOL)9,9,8
    8 WID=RAFT
    9 EFFW=3000.0/SQRT(FY)*2.0*TOUT
      BOUT=WID
      IF(WID-EFFW)11,11,12
   12 BOUT=EFFW
   11 EFFW=3000.0/SQRT(FY)*2.0*TIN
      BIN=WID
      IF(WID-EFFW)15,15,16
   16 BIN=EFFW
   15 CONTINUE
      AREA=WEB*(D-TOUT-TIN)+BOUT*TOUT+BIN*TIN
      CIN=(BOUT*TOUT*(D-TOUT/2.0)+WEB*(D-TOUT-TIN)*(TIN/2.0
     *+D/2.0-TOUT /2.0)+BIN*TIN**2/2.0)/AREA
      COUT=D-CIN
      ERT= BOUT*COUT**3/3.0-(BOUT-WEB)*(COUT-TOUT)**3/3.0 +
     *BIN*CIN**3/3.0-(BIN-WEB)*(CIN-TIN)**3/3.0
      SO= ERT/COUT
      SI= ERT/CIN
      RETURN
      END
```

```
      SUBROUTINE INPUT
      COMMON OPTL(6),A(9,17),OPSAV(10)
      COMMON PLTM(10),FLAN(10),GIRT(10),WC(6)
      COMMON WEBPL(6),WEB(10),NOCOL,NORAF,NOTOT
      COMMON KUTE,NOFLA,NOWEB,NPS,BAY,GD,NOGRT,H,SPAN,DL,VL
      COMMON WL,WFVP,FIRST,SPACE,NOCBM,KPLT,KWIND
      COMMON CBM(4,7),FLTI(10),FLTO(10),PLCOR(10)
      COMMON FY1,FY2,COST1,COST2,NOALT,KALT(10)
      COMMON TABLE(4,7),BRACE(5),ITLE(76),LANA,LDES,LPAT
      COMMON SCOS1,SCOS2,IMOT
      DIMENSION LIST(32),KLIST(6),VALUE(10),KLM(76)
      EQUIVALENCE (BD,A(3,2)),(COLU,A(6,2)),(ED,A(1,2))
      EQUIVALENCE (RD,A(4,2)),(RAFT,A(5,2)),(SLOPE,A(2,2))
      EQUIVALENCE(XBRE,A(7,2)),(DBRE,A(8,2)),(AOH,A(9,2))
C     LIST( 1) IS ANAL OR -1042955821
      LIST( 1)=-1042955821
C     LIST( 2) IS BASE OR -1027480891
      LIST( 2)=-1027480891
C     LIST( 3) IS BAY OR -1027479488
      LIST( 3)=-1027479488
C     LIST( 4) IS COLU OR -1009331228
      LIST( 4)=-1009331228
C     LIST( 5) IS COMB OR -1009331006
      LIST( 5)=-1009331006
C     LIST( 6) IS DEPT OR -993667101
      LIST( 6)= -993667101
C     LIST( 7) IS DESI OR -993664311
      LIST( 7)= -993664311
C     LIST( 8) IS EAVE OR -977148475
      LIST( 8)= -977148475
C     LIST( 9) IS EXEC OR -974666301
      LIST( 9)= -974666301
C     LIST(10) IS FLAN OR -959200811
      LIST(10)= -959200811
C     LIST(11) IS GIRT OR -943072797
      LIST(11)= -943072797
C     LIST(12) IS HEIG OR -926561849
      LIST(12)= -926561849
C     LIST(13) IS LOAD OR -740900412
      LIST(13)= -740900412
C     LIST(14) IS PART OR -675161629
      LIST(14)= -675161629
C     LIST(15) IS PEAK OR -674905646
      LIST(15)= -674905646
C     LIST(16) IS PLAT OR -673988125
      LIST(16)= -673988125
C     LIST(17) IS PURL OR -672867885
      LIST(17)= -672867885
C     LIST(18) IS RAFT OR -641612061
      LIST(18)= -641612061
C     LIST(19) IS SLOP OR -489433385
```

```
      LIST(19)= -489433385
C     LIST(20) IS SPAN OR -489176619
      LIST(20)= -489176619
C     LIST(21) IS WEB OR -423247296
      LIST(21)= -423247296
C     LIST(22) IS XBRE OR -406660667
      LIST(22)=-406660667
C     LIST(23) IS DBRE OR -993863227
      LIST(23)=-993863227
C     LIST(24) IS AOH OR -1042888640
      LIST(24)=-1042888640
C     LIST(25) IS STEE OR -488389179
      LIST(25)=-488389179
C     LIST(26) IS ALTE OR -1043078203
      LIST(26)=-1043078203
C     LIST(27) IS VARY OR -440280600
      LIST(27)=-440280600
C     LIST(28) IS BRAC OR -1025916477
      LIST(28)=-1025916477
C     LIST(29) IS TITL OR -473308205
      LIST(29)= -473308205
C     LIST(30) IS OUTP OR -689642537
      LIST(30)= -689642537
C     LIST(31) IS COST OR -1009327389
      LIST(31)= -1009327389
C     LIST(32) IS OMIT OR -690697757
      LIST(32)= -690697757
   50 READ(1,51)KTYPE,KLM
   51 FORMAT(A4,76A1)
      CALL SCAN (NO,VALUE,KLM)
      DO 53 I=1,32
      IF(KTYPE-LIST(I))53,54,53
   53 CONTINUE
      WRITE(3,52)KTYPE
   52 FORMAT(' ',A4,' IS AN INVALID CARD TYPE - CARD'
     ¤,' IGNORED')
      GO TO 50
   54 GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,
     ¤19,20,21,22, 23,24,25,26,27,28,29,30,31,32),I
    1 KUTE=1
      GO TO 50
    2 LINO=3
      GO TO 200
    3 BAY=VALUE(1)
      GO TO 50
    4 LINO=6
      GO TO 200
    5 NOCBM=NOCBM+1
      DO 120 I=1,7
  120 CBM(NOCBM,I)=VALUE(I)
      GO TO 50
```

```
    6 GD=VALUE(1)
      GO TO 50
    7 KUTE=0
      GO TO 50
    8 LINO=1
      GO TO 200
    9 RETURN
   10 NOFLA=NO
      DO 91 I=1,10
   91 FLAN(I)=VALUE(I)
      GO TO 50
   11 NOGRT=NO
      DO 92 I=1,10
   92 GIRT(I)=VALUE(I)
      GO TO 50
   12 H=VALUE(1)
      GO TO 50
   13 KLIST(1)=-993672764
C     KLIST( 1) IS DEAD OR -993672764
      KLIST(2)=-741743163
C     KLIST( 2) IS LIVE OR -741743163
      KLIST(3)=-422980156
C     KLIST( 3) IS WIND OR -422980156
      KLIST(4)=-1009330749
C     KLIST( 4) IS CONC OR -1009330749
      CALL LOOK(4,KLM,KLIST,4,KTYPE)
      GO TO (71,72,73,74,75),KTYPE
   71 DL=VALUE(1)
      GO TO 50
   72 VL=VALUE(1)
      GO TO 50
   73 KLIST(1)=-725429055
C     KLIST( 1) IS MBMA OR -725429055
      KLIST(2)=-1042103355
C     KLIST( 2) IS ASCE OR -1042103355
      KLIST( 3)= -673589551
C     KLIST( 3) IS PROJ OR -673589551
      KLIST(4)=-1042687551
C     KLIST( 4) IS ARRA OR -1042687551
      CALL LOOK(4,KLM,KLIST,4,KWIND)
      GO TO(81,82,83,84,90),KWIND
   81 KLIST(1)=-423172649
C     KLIST( 1) IS WFVP OR -423172649
      CALL LOOK(4,KLM,KLIST,1,LTYPE)
      GO TO (86,87),LTYPE
   87 WFVP=1.1
      GO TO 88
   86 WFVP=VALUE(2)
   88 WL=VALUE(1)
   82 GO TO 90
   83 WC(1)=1.0
```

```
      WC(2)=1.0
      WC(3)=0.0
      WC(4)=0.0
      WC(5)=0.0
      WC(6)=0.0
      GO TO 90
   84 WC(1)=VALUE(2)
      WC(2)=VALUE(3)
      WC(3)=VALUE(4)
      WC(4)=VALUE(5)
      WC(5)=VALUE(6)
      WC(6)=VALUE(7)
   90 WL=VALUE(1)
      GO TO 50
   74 N=VALUE(1)
      DO 215 I=1,5
  215 TABLE(N,I)=VALUE(I+1)
C     KLIST( 1) IS LEFT OR -742013213
      KLIST( 1)= -742013213
C     KLIST( 2) IS RIGH OR -641087544
      KLIST( 2)= -641087544
      CALL LOOK(4,KLM,KLIST,2,LEFT)
      GO TO (210,211,210),LEFT
  210 LEFT=-1
      GO TO 212
  211 LEFT=1
C     KLIST( 1) IS COLU OR -1009331228
  212 KLIST( 1)=-1009331228
C     KLIST( 2) IS RAFT OR -641612061
      KLIST( 2)= -641612061
      CALL LOOK (4,KLM,KLIST,2,LTYPE)
      GO TO (213,213,214),LTYPE
  214 LTYPE=LTYPE-2
  213 LTYPE=LTYPE+1
      TABLE(N,6)=LEFT+LTYPE
      GO TO 50
   75 WRITE(3,191)
  191 FORMAT(' INVALID OR NO TYPE OF LOAD - CARD IGNORED')
      GO TO 50
   14 NOCOL=VALUE(1)
      NORAF=VALUE(2)
      NORFA=VALUE(3)
      NOTOT=NOCOL+NORAF
      KPLT=1
      DO 121 I=1,NOTOT
      READ(1,122)KLM
  122 FORMAT(4X,76A1)
      CALL SCAN(NO,VALUE,KLM)
      PLCOR(I)=VALUE(1)
      FLTO(I)=VALUE(2)
      WEB(I)=VALUE(3)
```

```
  121 FLTI(I)=VALUE(4)
      GO TO 50
   15 LINO=4
      GO TO 200
   16 IF(NO-3)56,56,57
   57 NO=3
   56 NPS=NO
      DO 55 I=1,NO
      L=NO+1-I
   55 PLTM(I)=VALUE(L)
      GO TO 50
   17 FIRST=VALUE(1)
      SPACE=VALUE(2)
      GO TO 50
   18 LINO=5
      GO TO 200
   19 LINO=2
      GO TO 200
   20 SPAN=VALUE(1)
      GO TO 50
   21 NOWEB=NO
      DO 190 I=1,6
  190 WEBPL(I)=VALUE(I)
      GO TO 50
   22 LINO=7
      GO TO 200
   23 LINO=8
      GO TO 200
   24 LINO=9
      GO TO 200
   25 FY1=VALUE(1)
      COST1=VALUE(2)
      GO TO 50
   26 FY2=VALUE(1)
      COST2=VALUE(2)
      GO TO 50
   27 NOALT=NO
      DO 250 I=1,NO
  250 KALT(I)=VALUE(I)
      GO TO 50
  200 IF(NO-1) 201,202,201
  202 A(LINO,1)=0.0
      A(LINO,2)=VALUE(1)
      GO TO 50
  201 KLIST( 1)= -472498112
C     KLIST( 1) IS TO OR -472498112
      CALL LOOK(2,KLM,KLIST,1,LTYPE)
      GO TO (203,204),LTYPE
  204 A(LINO,1)=1.0
      A(LINO,2)=VALUE(1)
      A(LINO,3)=1.0
```

```
      A(LINO,4)=NO
      A(LINO,5)=1.0
      A(LINO,6)=NO
      DO 205 I=1,NO
  205 A(LINO,I+7)=VALUE(I)
      GO TO 50
  203 KLIST( 1)=-1024966592
C     KLIST( 1) IS BY OR -1024966592
      CALL LOOK(2,KLM,KLIST,1,LTYPE)
      GO TO (206,207),LTYPE
  207 A(LINO,1)=3.0
  208 A(LINO,3)=VALUE(1)
      A(LINO,2)=VALUE(1)
      A(LINO,4)=VALUE(2)
      A(LINO,5)=VALUE(1)
      A(LINO,6)=VALUE(2)
      GO TO 50
  206 A(LINO,1)=2.0
      A(LINO,7)=VALUE(3)
      GO TO 208
   28 DO 221 I=1,5
  221 BRACE(I)=VALUE(I)
      GO TO 50
   29 ITLE(1)=1
      DO 220 I=2,76
  220 ITLE(I)=KLM(I)
      GO TO 50
   30 LANA=0
      LDES=0
C     KLIST( 1) IS ANAL OR -1042955821
      KLIST( 1)=-1042955821
      CALL LOOK(4,KLM,KLIST,1,KTYPE)
      GO TO (222,223),KTYPE
  222 LANA=1
C     KLIST( 1) IS DESI OR -993664311
  223 KLIST( 1)= -993664311
      CALL LOOK(4,KLM,KLIST,1,KTYPE)
      GO TO (224,225),KTYPE
  224 LDES=1
C     KLIST( 1) IS PATH OR -675159096
  225 KLIST( 1)= -675159096
      CALL LOOK(4,KLM,KLIST,1,KTYPE)
      GO TO (226,227),KTYPE
  226 LPAT=1
  227 GO TO 50
   31 SCOS1=VALUE(1)
      SCOS2=VALUE(2)
      GO TO 50
   32 IMOT=99
      GO TO 50
      END
```

```
      SUBROUTINE LOCATE(FIRST,SPACE,PURL,NOPUR,SPAN,GD,
     1SLOPE,BRACE,MAST)
      DIMENSION PURL(20),MAST(20),BRACE(5)
      DO 20 I=1,20
   20 PURL(I)=0.0
      ALPHA=ATAN(SLOPE/12.0)
      PURL(1)=FIRST/12.0*COS(ALPHA)
      DO 1 I=2,21
      PURL(I)=PURL(I-1)+SPACE/12.0
      IF(PURL(I)-(SPAN/2.0-GD/12.0))1,1,2
    1 CONTINUE
      WRITE(3,3)
    3 FORMAT(' MAX NUMBER OF PURLINS EXCEEDED')
      STOP 2
    2 PURL(I-1)=PURL(I-1)-(PURL(I)-(SPAN/2.0-GD/12.0))/2.0
      PURL(I)=SPAN/2.0-GD/12.0
      NOPUR=I
      DO 4 I=1,20
    4 MAST(I)=I
      DO 5 I=1,5
      IF(BRACE(I))6,5,6
    6 DUM=ABS(BRACE(I)-PURL(1))
      IND=1
      DO 7 K=2,20
      IF(DUM-ABS(BRACE(I)-PURL(K)))7,7,8
    8 DUM=ABS(BRACE(I)-PURL(K))
      IND=K
    7 CONTINUE
      MAST(IND)=-IND
    5 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE LOOK (N,KLM,KLIST,M,KTYPE)
      DIMENSION KLM(76),KLIST(6),I(4)
      IM=77-N
      DO 12 II=1,IM
      DO 13 IL=1,4
      IK=II+IL-1
   13 I(IL)=KLM(IK)
      IF(N-1)2,1,2
    1 MERGE=I(1)
      GO TO 3
    2 MERGE=I(1)+12566463+I(2)/256
C    -12566463 IS EQUIVALENT TO BLANKS IN LOWER THREE BYTES
C    DIVISION BY 256 SHIFTS RIGHT ONE BYTE
      IF(N-2)4,3,4
    4 MERGE=MERGE+49087+I(3)/65536
C    -49087 IS EQUIVALENT TO BLANKS IN LOWER TWO BYTES
C    DIVISION BY 65536 SHIFTS RIGHT TWO BYTES
      IF(N-3) 5,3,5
    5 MERGE=MERGE+191+I(4)/16777216
C    -191 IS EQUIVALENT TO A BLANK IN THE LOWER BYTE
C    DIVISION BY 16777216 SHIFTS RIGHT THREE BYTES
      IF(N-4) 6,3,6
    6 WRITE(3,7)
    7 FORMAT(' N OUT OF RANGE  N=4 ASSUMED')
    3 DO 12 IK=1,M
      IF(KLIST(IK)-MERGE)12,16,12
   12 CONTINUE
      IK=M+1
   16 KTYPE=IK
      RETURN
      END
```

```
      SUBROUTINE NITIAL
      COMMON OPTL(6),A(9,17),OPSAV(10)
      COMMON PLTM(10),FLAN(10),GIRT(10),WC(6)
      COMMON WEBPL(6),WEB(10),NOCOL,NORAF,NOTOT
      COMMON KUTE,NOFLA,NOWEB,NPS,BAY,GD,NOGRT,H,SPAN,DL,VL
      COMMON WL,WFVP,FIRST,SPACE,NOCBM,KPLT,KWIND
      COMMON CBM(4,7),FLTI(10),FLTO(10),PLCOR(10)
      COMMON FY1,FY2,COST1,COST2,NOALT,KALT(10)
      COMMON TABLE(4,7),BRACE(5),ITLE(76),LANA,LDES,LPAT
      COMMON SCOS1,SCOS2,IMOT
      EQUIVALENCE (BD,A(3,2)),(COLU,A(6,2)),(ED,A(1,2))
      EQUIVALENCE (RD,A(4,2)),(RAFT,A(5,2)),(SLOPE,A(2,2))
      EQUIVALENCE(XBRE,A(7,2)),(DBRE,A(8,2)),(AOH,A(9,2))
      IMOT=0
      SCOS1=0.0
      SCOS2=0.0
      OPTL(1)=9.0
      DO 21 I=1,4
      DO 21 J=1,6
   21 TABLE(I,J)=0.0
      DO 301 I=1,5
  301 BRACE(I)=0.0
      KPLT=0
      DO 1 I=1,9
      DO 1 J=1,17
    1 A(I,J)=0.0
      ITLE(1)=0
      LANA=1
      LDES=1
      LPAT=0
      NOCBM=0
      DO 17 I=1,4
      DO 17 J=1,7
   17 CBM(I,J)=0.0
      NOALT=0
      DO 20 I=1,10
   20 KALT(I)=0
      AOH=99.9
      XBRE=0.0
      DBRE=0.0
      FY1=36000.0
      COST1=0.16
      FY2=0.0
      COST2=0.0
      WFVP=1.1
      NPS=3
      PLTM(1)=16.0
      PLTM(2)=11.0
      PLTM(3)=8.0
      SLOPE=2.0
      FIRST=11.0
```

```
      SPACE=60.0
      FLAN(1)=0.1875
      DO 2 I=2,8
    2 FLAN(I)=FLAN(I-1)+0.0625
      NOFLA=8
      NOWEB=5
      WEBPL(1)=0.125
      DO 3 I=2,5
    3 WEBPL(I)=WEBPL(I-1)+0.0625
      DO 4 I=1,10
      FLTI(I)=0.375
      FLTO(I)=0.375
    4 WEB(I)=0.1875
      RD=0.0
      GD=7.5
      KWIND=1
      WL=25.0
      VL=20.0
      DL=4.0
      A(3,1)=1.0
      A(3,2)=7.0
      A(3,3)=1.0
      A(3,4)=3.0
      A(3,8)=7.0
      A(3,9)=9.0
      A(3,10)=11.0
      A(4,1)=0.0
      A(2,1)=0.0
      A(1,1)=-1.0
      A(5,1)=-1.0
      A(6,1)=-1.0
      BAY=20.0
      NOGRT=-1
      CALL INPUT
      IF(RD)30,31,30
   31 IF(SLOPE-1.5)33,32,32
   32 RD=6.0
      GO TO 30
   33 DO 34 I=1,17
   34 A(4,I)=A(1,I)
   30 GO TO (22,23,24,24,22),KWIND
   22 CALL WIND(SLOPE,H,SPAN,WC)
      GO TO 24
C     ASCE WIND LOAD COEFFICIENT ROUTINE
   23 WL=20.0
      WC(1)=1.0
      WC(4)=-0.45
      WC(5)=-0.45
      WC(6)=0.0
      THETA=ATAN(SLOPE/12.0)*180.0/3.1415926
      IF(THETA-20.0)25,26,26
```

```
25 WC(2)=-0.6
   GO TO 30
26 IF(THETA-30.0)27,28,28
27 WC(2)=(1.2*THETA-36.0)/20.0
   GO TO 30
28 IF(THETA-60.0)29,31,31
29 WC(2)=(0.3*THETA-9.0)/20.0
   GO TO 30
31 WC(2)=-0.45
30 WC(3)=WC(2)
24 IF(NOCBM)14,15,14
15 NOCBM=2
   CBM(1,1)=1.0
   CBM(1,2)=1.0
   CBM(2,1)=0.75
   CBM(2,3)=0.75
14 IF(A(1,1))5,6,6
 5 A(1,1)=3.0
   A(1,2)=0.3*SPAN
   A(1,3)=0.3*SPAN
   A(1,4)=0.4*SPAN
 6 IF(A(5,1))7,8,8
 7 A(5,1)=1.0
   A(5,3)=1.0
   A(5,4)=6.0
   A(5,2)=4.0
   A(5,8)=5.0
   A(5,9)=6.0
   A(5,10)=7.5
   A(5,11)=9.0
   A(5,12)=10.0
   A(5,13)=12.0
 8 IF(A(6,1))9,10,10
 9 A(6,1)=1.0
   A(6,4)=6.0
   A(6,2)=4.0
   A(6,3)=1.0
   A(6,8)=5.0
   A(6,9)=6.0
   A(6,10)=7.5
   A(6,11)=9.0
   A(6,12)=10.0
   A(6,13)=12.0
10 IF(NOGRT)11,12,12
11 GIRT(1)=0.5
   GIRT(2)=3.0
   GIRT(3)=8.0
   NUM=(H-8.0)/4.0+0.5
   SPA=(H-8.0)/(NUM)
   DO 13 I=1,NUM
13 GIRT(I+3)=8.0+SPA*I
```

```
      NOGRT=NUM+3
   12 RETURN
      END
```

```
      SUBROUTINE OPTMUM (KIN)
      COMMON OPTL(6),A(9,17),OPSAV(10)
      GO TO (1,2,3),KIN
    1 DO 100 I=1,10
  100 OPSAV(I)=9.9E50
      OPTL(4)=1.0
      NO=OPTL(1)
      DO 300 I=1,NO
      A(I,5)=A(I,3)
  300 A(I,6)=A(I,4)
      OPTL(5)=9.9E50
      OPTL(6)=9.9E50
      DO 1101 LINE=1,NO
      ITYPE=A(LINE,1)+1.0
      KCODE=1
      GO TO (101,102,103,104),ITYPE
  101 GO TO (1101,1102),KCODE
 1101 CONTINUE
      GO TO 111
  102 NOVAL=A(LINE,4)
      NOLD=1
      NEW=1
  107 IF(NEW+NOLD-NOVAL)105,105,106
  105 NEW=NEW+NOLD
      NOLD=NEW-NOLD
      GO TO 107
  106 A(LINE,2)=A(LINE,NEW+7)
      GO TO 101
  103 NOVAL=(A(LINE,4)-A(LINE,3))/A(LINE,7)+1.5
      NOLD=1
      NEW=1
  108 IF(NEW+NOLD-NOVAL)109,109,110
  109 NEW=NEW+NOLD
      NOLD=NEW-NOLD
      GO TO 108
  110 A(LINE,2)=A(LINE,3)+(NEW-1)*A(LINE,7)
      GO TO 101
  104 GOLD=(-1.0+SQRT(5.0))/2.0
      A(LINE,2)=(A(LINE,4)-A(LINE,3))*GOLD+A(LINE,3)
      GO TO 101
  111 DO 112 LINE=1,NO
      IF(A(LINE,1))113,112,113
  112 CONTINUE
      KIN=4
      RETURN
  113 OPTL(2)=A(LINE,2)
      KIN=2
      OPTL(4)=LINE
      RETURN
    2 LINE=OPTL(4)
      OPTL(6)=OPTL(3)
```

```
      OPTL(5)=OPTL(2)
      ITYPE=A(LINE,1)
      GO TO (5,6,7),ITYPE
    5 NOVAL=A(LINE,4)
      NOLD=1
      NEW=1
  114 IF(NEW+NOLD-NOVAL)115,115,116
  115 NEW=NEW+NOLD
      NOLD=NEW-NOLD
      GO TO 114
  116 A(LINE,2)=A(LINE,NOLD+7)
  121 OPTL(2)=A(LINE,2)
      KIN=3
      RETURN
    6 NOVAL=(A(LINE,4)-A(LINE,3))/A(LINE,7)+1.5
      NOLD=1
      NEW=1
  118 IF(NEW+NOLD-NOVAL)119,119,120
  119 NEW=NEW+NOLD
      NOLD=NEW-NOLD
      GO TO 118
  120 A(LINE,2)=A(LINE,3)+(NOLD-1)*A(LINE,7)
      GO TO 121
    7 GOLD=1.0-(-1.0+SQRT(5.0))/2.0
      A(LINE,2)=(A(LINE,4)-A(LINE,3))*GOLD+A(LINE,3)
      GO TO 121
    3 LINE=OPTL(4)
      ITYPE=A(LINE,1)
  235 COMP=OPTL(6)-OPTL(3)
      IF(COMP)231,230,231
  230 IF(OPTL(5)-OPTL(2))232,232,233
  233 OPTL(3)=OPTL(3)+1.0
      GO TO 235
  232 OPTL(6)=OPTL(6)+1.0
      GO TO 235
  231 LEFT=1
      IF(OPTL(5)-OPTL(2))122,123,123
  122 LEFT=2
  123 GO TO (131,132,133),ITYPE
  131 N=A(LINE,4)
      DO 135 I=1,N
      IF(OPTL(2)-A(LINE,I+7))135,136,135
  135 CONTINUE
      STOP 135
  136 MNEW=I
      DO 137 I=1,N
      IF(OPTL(5)-A(LINE,I+7))137,138,137
  137 CONTINUE
      STOP 137
  138 MOLD=I
      IF(A(LINE,6)-A(LINE,5)-1.5)140,139,139
```

```
  139 IF(COMP)141,142,142
  142 GO TO (143,144),LEFT
  143 A(LINE,6)=MOLD-1
      OPTL(6)=OPTL(3)
      OPTL(5)=OPTL(2)
  145 NOVAL=A(LINE,6)-A(LINE,5)+1.5
  157 NOLD=1
      NEW=1
  148 IF(NEW+NOLD-NOVAL) 146,146,147
  146 NEW=NEW+NOLD
      NOLD=NEW-NOLD
      GO TO 148
  147 KDUM=A(LINE,5)
      KDUM=KDUM+NOLD+6
      A(LINE,2)=A(LINE,KDUM)
  150 OPTL(2)=A(LINE,2)
      KIN=3
      RETURN
  144 A(LINE,5)=MOLD+1
      OPTL(6)=OPTL(3)
      OPTL(5)=OPTL(2)
  158 NOVAL=A(LINE,6)-A(LINE,5)+1.5
      NOLD=1
      NEW=1
  154 IF(NEW+NOLD-NOVAL)152,152,153
  152 NEW=NEW+NOLD
      NOLD=NEW-NOLD
      GO TO 154
  153 KDUM=A(LINE,5)
      KDUM=KDUM+NEW+6
      A(LINE,2)=A(LINE,KDUM)
      GO TO 150
  141 GO TO (155,156),LEFT
  156 A(LINE,6)=MNEW-1
      GO TO 145
  155 A(LINE,5)=MNEW+1
      GO TO 158
  132 IF(A(LINE,6)-A(LINE,5)-A(LINE,7))140,140,160
  160 IF(COMP)161,161,162
  162 GO TO (163,164),LEFT
  163 A(LINE,6)=OPTL(5)-A(LINE,7)
      OPTL(5)=OPTL(2)
      OPTL(6)=OPTL(3)
  165 NOVAL=(A(LINE,6)-A(LINE,5))/A(LINE,7)+1.5
      NEW=1
      NOLD=1
  166 IF(NEW+NOLD-NOVAL)167,167,1168
  167 NEW=NEW+NOLD
      NOLD=NEW-NOLD
      GO TO 166
 1168 A(LINE,2)=A(LINE,5)+(NOLD-1)*A(LINE,7)
```

```
 168 OPTL(2)=A(LINE,2)
     KIN=3
     RETURN
 164 A(LINE,5)=OPTL(5)+A(LINE,7)
     OPTL(5)=OPTL(2)
     OPTL(6)=OPTL(3)
 170 NOVAL=(A(LINE,6)-A(LINE,5))/A(LINE,7)+1.5
     NEW=1
     NOLD=1
 171 IF(NEW+NOLD-NOVAL)172,172,173
 172 NEW=NEW+NOLD
     NOLD=NEW-NOLD
     GO TO 171
 173 A(LINE,2)=A(LINE,5)+(NEW-1)*A(LINE,7)
     GO TO 168
 161 GO TO (180,181),LEFT
 181 A(LINE,6)=OPTL(2)-A(LINE,7)
     GO TO 165
 180 A(LINE,5)=OPTL(2)+A(LINE,7)
     GO TO 170
 133 IF((A(LINE,4)-A(LINE,3))/6.0-(A(LINE,6)-A(LINE,5)))
    □1133,140,140
1133 IF(COMP) 190,190,191
 191 GO TO (192,193),LEFT
 192 A(LINE,6)=OPTL(5)
     OPTL(5)=OPTL(2)
     OPTL(6)=OPTL(3)
 194 GOLD=1.0-(-1.0+SQRT(5.0))/2.0
 197 A(LINE,2)=A(LINE,5)+GOLD*(A(LINE,6)-A(LINE,5))
 195 OPTL(2)=A(LINE,2)
     KIN=3
     RETURN
 193 A(LINE,5)=OPTL(5)
     OPTL(5)=OPTL(2)
     OPTL(6)=OPTL(3)
 196 GOLD=(-1.0+SQRT(5.0))/2.0
     GO TO 197
 190 GO TO (198,199),LEFT
 199 A(LINE,6)=OPTL(2)
     GO TO 194
 198 A(LINE,5)=OPTL(2)
     GO TO 196
 140 A(LINE,2)=OPTL(5)
     IF(COMP)201,201,200
 200 A(LINE,2)=OPTL(2)
 201 N=OPTL(1)
 209 LINE=LINE+1
     IF(LINE-N)2209,2209,205
2209 DO 204 I=LINE,N
     IF(A(I,1))203,204,203
 204 CONTINUE
```

```
      LINE=N
      GO TO 205
  203 KCODE=2
      LINE=I
      ITYPE=A(LINE,1)
      GO TO (102,103,104),ITYPE
 1102 KIN=2
      A(LINE,6)=A(LINE,4)
      A(LINE,5)=A(LINE,3)
      OPTL(4)=LINE
      OPTL(2)=A(LINE,2)
      RETURN
  205 DO 206 I=1,N
      IF(OPSAV(I)-A(I,2))207,206,207
  206 CONTINUE
      KIN=4
      RETURN
  207 DO 208 I=1,N
  208 OPSAV(I)=A(I,2)
      OPSAV(10)=OPTL(6)
      IF(COMP) 212,212,211
  211 OPSAV(10)=OPTL(3)
  212 LINE=0
      GO TO 209
      END
```

```
      SUBROUTINE OUTPUT(IPAGE,ITLE,AXL,SHL,SML,AXR,SHR,SMR,
     □NOTOT,L, D,SO,SI,AREA,FLTO,FLTI,WEB,WIDI,WIDO,PARTL)
      DIMENSION AXL(10,7),SHL(10,7),SML(10,7),AXR(10,7),
     □SHR(10,7), SMR(10,7),D(10,7),SO(10,7),SI(10,7),
     □AREA(10,7),FLTO(10), FLTI(10),WEB(10),ITLE(51),
     □WIDI(10),WIDO(10),PARTL(10)
  108 FORMAT(' ')
  106 FORMAT(I1,50A1,'    PAGE',I3)
  101 FORMAT('0',T17,'LEFT SIDE',T43,'RIGHT SIDE'/ T24,
     □'BENDING',T50,'BENDING'/ ' PART SEC AXIAL SHEAR OUTER'
     □,' INNER   AXIAL SHEAR OUTER INNER')
  107 FORMAT(2I4,-3PF7.1,3F6.1,F8.1,3F6.1)
  127 FORMAT('OCOMBINATION NO.',I2,' - STRESSES')
      ICTL=5
      DO 110 I=1,NOTOT
      TAN=(D(I,1)-D(I,7))/12.0/PARTL(I)
      SECT=1.0/COS(ATAN(TAN))
      IF(ICTL-5)111,112,111
  112 ICTL=0
      IPAGE=IPAGE+1
      WRITE(3,106)(ITLE(M),M=1,51),IPAGE
      WRITE(3,127)L
      WRITE(3,101)
  111 ICTL=ICTL+1
      DO 109 J=1,7
      AL=AXL(I,J)/AREA(I,J)
      VTL=SML(I,J)/SI(I,J)*FLTI(I)*WIDI(I)*TAN
      VCOR=SHL(I,J)+VTL
      SL=VCOR /(WEB(I)*(D(I,J)-FLTO(I)-FLTI(I)))
      BOL=-SML(I,J)*12.0/SO(I,J)
      BIL=SML(I,J)*12.0/SI(I,J)*SECT
      AR=AXR(I,J)/AREA(I,J)
      VTR=SMR(I,J)/SI(I,J)*FLTI(I)*WIDI(I)*TAN
      VCOR=SHR(I,J)+VTR
      SR=VCOR /(WEB(I)*(D(I,J)-FLTO(I)-FLTI(I)))
      BOR=-SMR(I,J)*12.0/SO(I,J)
      BIR=SMR(I,J)*12.0/SI(I,J)*SECT
  109 WRITE(3,107)I,J,AL,SL,BOL,BIL,AR,SR,BOR,BIR
  110 WRITE(3,108)
      RETURN
      END
```

```
      SUBROUTINE PARTS (REFL,NPS,PLTM,NHOLD)
      DIMENSION PLTM(10),NHOLD(10),N(10),WL(10)
      HOLD=999.99
      NPSM1=NPS-1
      WL(1)=REFL
      J=1
   10 DO 1 K=J,NPSM1
    1 N(K+1)=0
   13 N(J)=WL(J)/PLTM(J)
      IF(N(J)*PLTM(J)-WL(J))2,3,3
    2 N(J)=N(J)+1
    3 TOTL=0.0
      DO 4 K=1,NPS
    4 TOTL=TOTL+N(K)*PLTM(K)
      IF(TOTL-HOLD)5,6,6
    5 HOLD=TOTL
      DO 7 K=1,NPS
    7 NHOLD(K)=N(K)
    6 IF(J-NPSM1) 9,8,8
    9 N(J)=N(J)-1
      J=J+1
      WL(J)=REFL
      JM1=J-1
      DO 20 K=1,JM1
   20 WL(J)=WL(J)-N(K)*PLTM(K)
      GO TO 10
    8 IF(N(NPSM1))11,12,11
   11 N(NPSM1)=N(NPSM1)-1
      WL(NPS)=REFL
      DO 22 K=1,NPSM1
   22 WL(NPS)=WL(NPS)-N(K)*PLTM(K)
      J=NPS
      GO TO 13
   12 J=J-1
      IF(J)15,16,15
   15 IF(N(J))14,12,14
   14 N(J)=N(J)-1
      WL(J+1)=REFL
      DO 21 K=1,J
   21 WL(J+1)=WL(J+1)-N(K)*PLTM(K)
      J=J+1
      GO TO 10
   16 RETURN
      END
```

```
      SUBROUTINE PRINT(IPAGE,ITLE,KLN,AXL,SHL,SML,AXR,SHR,
     ¤SMR,NOTOT,L,R)
      DIMENSION AXL(10,7),SHL(10,7),SML(10,7),AXR(10,7),
     ¤SHR(10,7), SMR(10,7),ITLE(51),R(4)
  108 FORMAT(' ')
  106 FORMAT(I1,50A1,'   PAGE',I3)
  107 FORMAT(2I4,-3PF8.2,F7.2,F10.2,F9.2,F7.2,F10.2)
  101 FORMAT('0',T17,'LEFT SIDE',T43,'RIGHT SIDE'/ ' PART'
     ¤,' SEC  AXIAL  SHEAR   MOMENT   AXIAL  SHEAR   '
     ¤,' MOMENT')
  102 FORMAT('0DEAD LOAD - FORCES')
  103 FORMAT('0LIVE LOAD - FORCES')
  104 FORMAT('0WIND LOAD - FORCES')
  105 FORMAT('0CONCENTRATED LOAD NO.',I2,' - FORCES')
  127 FORMAT('0COMBINATION NO.',I2,' - FORCES')
      ICTL=5
      DO 110 I=1,NOTOT
      IF(ICTL-5)111,112,111
  112 ICTL=0
      IPAGE=IPAGE+1
      WRITE(3,106)(ITLE(M),M=1,51),IPAGE
      GO TO (121,122,123,124,126),KLN
  121 WRITE(3,102)
      GO TO 125
  122 WRITE(3,103)
      GO TO 125
  123 WRITE(3,104)
      GO TO 125
  124 WRITE(3,105)L
      GO TO 125
  126 WRITE(3,127)L
  125 WRITE(3,101)
  111 ICTL=ICTL+1
      DO 109 J=1,7
  109 WRITE(3,107)I,J,AXL(I,J),SHL(I,J),SML(I,J),AXR(I,J),
     ¤SHR(I,J), SMR(I,J)
  110 WRITE(3,108)
      WRITE(3,130)R
  130 FORMAT('           REACTIONS'/T12,'    LEFT   RIGHT'/
     ¤' VERTICAL  ',-3P2F8.2/' HORIZONTAL',2F8.2)
      RETURN
      END
```

```
      SUBROUTINE SCAN (NO,VALUE,KLM)
      DIMENSION VALUE(10),KLM(76)
      DO 1 I=1,10
    1 VALUE(I)=0.0
      NCOL=1
      N=1
      KPT=0
    2 IF(KLM(NCOL)-1614823488)4,3,4
C     -1614823488 IS EQUIVALENT TO A MINUS SIGN
    3 SGN=-1.0
      GO TO 5
    4 IF(KLM(NCOL)-1312833600)6,7,6
C     -1312833600 IS EQUIVALENT TO A PLUS SIGN
    7 SGN=1.0
    5 VALUE(N)=0.0
      GO TO 8
    6 IF(KLM(NCOL)-1262501952)9,10,9
C     -1262501952 IS EQUIVALENT TO A DECIMAL POINT
   10 KPT=1
      GO TO 7
    9 K=0
      ICOMP=-264224704
C     -264224704 IS EQUIVALENT TO A ZERO
   11 IF(KLM(NCOL)-ICOMP)12,13,12
   12 ICOMP=ICOMP+16777216
C     +16777216 IS EQUIVALENT TO THE DIFFERENCE
C                BETWEEN TWO DIGITS
      K=K+1
      IF(K-10)11,14,14
   14 NCOL=NCOL+1
   24 IF(NCOL-77)2,16,16
   16 NO=N-1
      RETURN
   13 SGN=1.0
      VALUE(N)=K
    8 NCOL=NCOL+1
      IF(NCOL-77)17,18,18
   17 IF(KLM(NCOL)-1262501952)20,19,20
C     -1262501952 IS EQUIVALENT TO A DECIMAL POINT
   19 KPT=1
      GO TO 8
   20 K=0
      ICOMP=-264224704
C     -264224704 IS EQUIVALENT TO A ZERO
   21 IF(KLM(NCOL)-ICOMP)22,23,22
   22 ICOMP=ICOMP+16777216
C     +16777216 IS EQUIVALENT TO THE DIFFERENCE
C                BETWEEN TWO DIGITS
      K=K+1
      IF(K-10)21,18,18
   18 VALUE(N)=VALUE(N)*SGN
```

```
      N=N+1
      KPT=0
      GO TO 24
   23 IF(KPT)25,26,25
   26 VALUE(N)=VALUE(N)*10.0+K
      GO TO 8
   25 VALUE(N)=VALUE(N)+K*10.0**(-KPT)
      KPT=KPT+1
      GO TO 8
      END
```

```
      SUBROUTINE SEARCH (SWTCH)
      COMMON OPTL(6),A(9,17),OPSAV(10)
      DIMENSION FFOT(9),TRY(9)
      IF(SWTCH)2,1,2
    1 SWTCH=1.0
      VECTR=0.0
      NO=OPTL(1)
      DO 3 I=1,NO
    3 FFOT(I)=1.0
    4 SUM=0.0
      DO 5 I=1,NO
      IF(FFOT(I))6,5,6
    6 SUM=SUM+1.0
    5 CONTINUE
      IF(SUM-2.0)7,8,8
    8 DO 9 I=1,NO
      IF(FFOT(I))10,11,10
   11 TRY(I)=OPSAV(I)
      GO TO 9
   10 IF(A(I,1))13,12,13
   13 ITYPE=A(I,1)
      IF(FFOT(I))14,16,15
   16 STOP 16
   14 GO TO (17,18,19),ITYPE
   17 DO 20 K=1,10
      IF(A(I,K+7)-OPSAV(I))20,21,20
   20 CONTINUE
      STOP 21
   21 IF(K-1)22,12,22
   22 TRY(I)=A(I,K+6)
      GO TO 9
   18 TRY(I)=OPSAV(I)-A(I,7)
   23 IF(TRY(I)-A(I,3))12,9,9
   19 TRY(I)=OPSAV(I)-(A(I,4)-A(I,3))*0.05
      GO TO 23
   15 GO TO (24,25,26),ITYPE
   24 DO 27 K=1,10
      IF(A(I,K+7)-OPSAV(I)) 27,28,27
   27 CONTINUE
      STOP 28
   28 KEND=A(I,4)
      IF(K-KEND)29,12,12
   29 TRY(I)=A(I,K+8)
      GO TO 9
   25 TRY(I)=OPSAV(I)+A(I,7)
   30 IF(TRY(I)-A(I,4))9,9,12
   26 TRY(I)=OPSAV(I)+(A(I,4)-A(I,3))*0.05
      GO TO 30
    9 CONTINUE
      DO 31 I=1,NO
   31 A(I,2)=TRY(I)
```

```
      RETURN
   12 IF(VECTR)32,7,32
   32 SWTCH=0.0
      DO 35 I=1,NO
   35 A(I,2)=OPSAV(I)
      RETURN
    2 IF(OPTL(3)-OPSAV(10))33,12,12
   33 DO 36 I=1,NO
   36 OPSAV(I)=A(I,2)
      OPSAV(10)=OPTL(3)
      VECTR=1.0
      GO TO 8
    7 I=1
   44 IF(FFOT(I))39,40,41
   41 FFOT(I)=-1.0
      GO TO 4
   39 FFOT(I)=0.0
      GO TO 4
   40 FFOT(I)=1.0
      IF(I-NO)43,42,43
   43 I=I+1
      GO TO 44
   42 SWTCH=-1.0
      RETURN
      END
```

```
      SUBROUTINE SELECT(NOFLA,NOWEB,FLAN,WEBPL,WEB,NOCBM,
     □FLTI,FLTO,D, ERT,AREA,SO,SI,SHEAR,OMENT,AXIAL,UBLO,
     □UBLI,AOH,FY,I, PARTL,WIDI,WIDO,KASTR,NOCOL)
      DIMENSION FLAN(10),WEBPL(6),WEB(10),FLTI(10),FLTO(10),
     □D(10,7), ERT(10,7),AREA(10,7),SO(10,7),SI(10,7),
     □SHEAR(8,10,7),OMENT(8,10,7),AXIAL(8,10,7),UBLO(10,7),
     □UBLI(10,7),PARTL(10),WIDI(10),WIDO(10)
      KWSAV=0
      KFOSV=0
      KFISV=0
      M=2*NOCBM
      CALL FINDER (WEB(I),WEBPL,NOWEB,KWEB)
      TAN=(D(I,1)-D(I,7))/12.0/PARTL(I)
      SECT=1.0/COS(ATAN(TAN))
   33 LEAD=1
    9 RATW=0.0
      DO 3 J=1,7
      CALL SSTRES(AOH,D(I,J),FLTI(I),FLTO(I),WEB(I),FY,FV)
      DO 3 K=1,M
      VT=OMENT(K,I,J)/SI(I,J)*FLTI(I)*WIDI(I)*TAN
      VCOR=SHEAR(K,I,J)+VT
      FVA=VCOR/((D(I,J)-FLTO(I)-FLTI(I))*WEB(I))
      DUM=ABS(FVA/FV)
      IF(DUM-1.0)2,2,6
    2 IF(DUM-RATW)3,3,4
    4 RATW=DUM
    3 CONTINUE
    5 IF(LEAD-2)10,100,10
   10 IF(KWEB-1)11,100,11
   11 KWEB=KWEB-1
      WEB(I)=WEBPL(KWEB)
  122 DO 12 J=1,7
   12 CALL INERT(D(I,J),FLTO(I),WEB(I),FLTI(I),WIDO(I),
     □WIDI(I), AREA(I,J),SO(I,J),SI(I,J),ERT(I,J),NOCOL,FY,
     □I)
      GO TO 9
    6 IF(KWEB-NOWEB)7,8,8
    7 KWEB=KWEB+1
      WEB(I)=WEBPL(KWEB)
      LEAD=2
      GO TO 122
    8 KASTR=999
      KWEB=NOWEB
  100 CONTINUE
      IF(KWSAV-KWEB)14,900,70
   70 WEB(I)=WEBPL(KWSAV)
      GO TO 900
   14 KWSAV=KWEB
   64 LEAD=1
      CALL FINDER(FLTO(I),FLAN,NOFLA,KFO)
   30 RATFO=0.0
```

```
      DO 13 J=1,7
      CALL STRESS(UBLO(I,J),UBLI(I,J),D(I,J),FLTO(I),FLTI(I)
     ¤,WIDI(I), WIDO(I),WEB(I),FBO,FBI,FA,FT,FY)
      DO 13 K=1,M
      SBO=OMENT(K,I,J)/SO(I,J)*(-12.0)
      SA=AXIAL(K,I,J)/AREA(I,J)
      IF(SBO)15,16,16
   15 DUM=SBO/FBO
      GO TO 17
   16 DUM=SBO/FT
   17 IF(SA)18,19,19
   18 DUM=DUM+SA/FA
      GO TO 20
   19 DUM=DUM+SA/FT
   20 IF(ABS(DUM)-RATFO)21,21,22
   22 RATFO=ABS(DUM)
   21 CONTINUE
      IF(ABS(DUM)-1.0)13,13,23
   13 CONTINUE
      IF(LEAD-2)24,25,24
   24 IF(KFO-1)26,25,26
   26 KFO=KFO-1
      FLTO(I)=FLAN(KFO)
      GO TO 27
   23 IF(KFO-NOFLA)28,29,29
   29 KASTR=999
      KFO=NOFLA
      GO TO 25
   28 KFO=KFO+1
      LEAD=2
      FLTO(I)=FLAN(KFO)
   27 DO 31 J=1,7
   31 CALL INERT(D(I,J),FLTO(I),WEB(I),FLTI(I),WIDO(I),
     ¤WIDI(I), AREA(I,J),SO(I,J),SI(I,J),ERT(I,J),NOCOL,FY,
     ¤I)
      GO TO 30
   25 CONTINUE
      IF(KFO-KFOSV)32,33,32
   32 KFOSV=KFO
      LEAD=1
      CALL FINDER(FLTI(I),FLAN,NOFLA,KFI)
   60 RATFI=0.0
      DO 43 J=1,7
      CALL STRESS(UBLO(I,J),UBLI(I,J),D(I,J),FLTO(I),FLTI(I)
     ¤,WIDI(I), WIDO(I),WEB(I),FBO,FBI,FA,FT,FY)
      DO 43 K=1,M
      SBI=OMENT(K,I,J)/SI(I,J)*12.0*SECT
      SA=AXIAL(K,I,J)/AREA(I,J)
      IF(SBI)45,46,46
   45 DUM=SBI/FBI
      GO TO 47
```

```
 46 DUM=SBI/FT
 47 IF(SA)48,49,49
 48 DUM=DUM+SA/FA
    GO TO 50
 49 DUM=DUM+SA/FT
 50 IF(ABS(DUM)-RATFI)51,51,52
 52 RATFI=ABS(DUM)
 51 CONTINUE
    IF(ABS(DUM)-1.0)43,43,53
 43 CONTINUE
    IF(LEAD-2)54,55,54
 54 IF(KFI-1)56,55,56
 56 KFI=KFI-1
    FLTI(I)=FLAN(KFI)
    GO TO 57
 53 IF(KFI-NOFLA)58,59,59
 59 KASTR=999
    KFI=NOFLA
    GO TO 55
 58 KFI=KFI+1
    LEAD=2
    FLTI(I)=FLAN(KFI)
 57 DO 61 J=1,7
 61 CALL INERT(D(I,J),FLTO(I),WEB(I),FLTI(I),WIDO(I),
   ¤WIDI(I), AREA(I,J),SO(I,J),SI(I,J),ERT(I,J),NOCOL,FY,
   ¤I)
    GO TO 60
 55 CONTINUE
    IF(KFI-KFISV)62,33,62
 62 KFISV=KFI
    GO TO 64
900 RETURN
    END
```

```
      SUBROUTINE SSTRES(AOH,D,TFO,TFI,W,FY,FV)
      HOT=(D-TFO-TFI)/W
      IF(AOH-1.0)1,1,2
    1 FK=4.0+5.34/AOH**2
      GO TO 3
    2 IF(AOH-3.0)4,5,5
    4 FK=5.34+4.0/AOH**2
      GO TO 3
    5 FK=5.34
    3 CV=45000000.0*FK/FY/HOT**2
      IF(CV-0.8)6,6,7
    7 CV=6000.0*SQRT(FK/FY)/HOT
    6 IF(AOH-3.0)8,11,11
    8 IF(CV-1.0)9,11,11
    9 FV=FY/2.89*(CV+(1.0-CV)/(1.15*SQRT(1.0+AOH**2)))
      GO TO 10
   11 FV=FY*CV/2.89
   10 IF(FV-0.4*FY)12,12,13
   13 FV=0.4*FY
   12 RETURN
      END
```

```
      SUBROUTINE STFNER (AOH,FLTI,FLTO,D,WEB,FY1,FY2,SHL,A,
     ¤NOSTR,STIFF, NOTOT,NOCOL,X,Y,ALPHA,NOALT,KALT,NOSTC,
     ¤XBRE)
      DIMENSION FLTI(10),FLTO(10),D(10,7),WEB(10),SHL(10,7),
     ¤A(10,7), STIFF(2,10),X(10,7),Y(10,7),KALT(10)
      DO 100 I=1,NOTOT
      FY=FY1
      DO 3 M=1,NOALT
      IF(I-KALT(M))3,2,3
    3 CONTINUE
      GO TO 66
    2 FY=FY2
   66 DO 100 J=1,7
      ASH=99.9
      HWEB=(D(I,J)-FLTO(I)-FLTI(I))
      SV=SHL(I,J)/(HWEB*WEB(I))
      CALL SSTRES(ASH,D(I,J),FLTO(I),FLTI(I),WEB(I),FY,FV)
      IF(FV-SV)5,6,6
    6 A(I,J)=9999.9
      GO TO 100
    5 ASH=3.0
      FACT=1.0
    7 CALL SSTRES(ASH,D(I,J),FLTO(I),FLTI(I),WEB(I),FY,FV)
      IF(FV-SV)9,8,8
    8 IF(ABS(ASH-3.0)-0.001)11,11,10
   11 A(I,J)=ASH*HWEB
      GO TO 100
   10 IF(ABS(FACT-0.01)-0.001)11,11,12
   12 ASH=ASH+FACT
      FACT=FACT/10.0
    9 ASH=ASH-FACT
      IF(ASH)7,12,7
  100 CONTINUE
      KK=1
      NOSTC=0
      DO 200 I=1,NOCOL
      DO 200 M=1,7
      J=7-M+1
      IF(A(I,J)-9999.0)20,200,200
   20 IF(NOSTC)21,22,21
   22 NOSTC=1
      STIFF(KK,1)=Y(I,J)
      STIFF(KK,2)=Y(I,J)-A(I,J)/12.0
      IF(STIFF(KK,2))200,200,23
   23 NOSTC=2
      GO TO 200
   21 IF(STIFF(KK,NOSTC)-Y(I,J))24,200,25
   24 IF(12.0*(STIFF(KK,NOSTC-1)-STIFF(KK,NOSTC))-A(I,J))
     ¤200,200,26
   26 STIFF(KK,NOSTC)=STIFF(KK,NOSTC-1)-A(I,J)/12.0
      IF(STIFF(KK,NOSTC)-Y(I,J))200,200,27
```

```
 27 STIFF(KK,NOSTC)=Y(I,J)
    GO TO 200
 25 STIFF(KK,NOSTC+1)=STIFF(KK,NOSTC)-A(I,J)/12.0
    IF(STIFF(KK,NOSTC+1))200,200,28
 28 NOSTC=NOSTC+1
200 CONTINUE
    IF(XBRE)30,31,30
 31 BRAKE=X(NOTOT,7)
    GO TO 32
 30 BRAKE=XBRE
 32 KK=2
    NOSTR=0
    LL=NOCOL+1
    DO 300 I=LL,NOTOT
    DO 300 J=1,7
    IF(X(I,J)-BRAKE)35,34,35
 34 BRAKE=X(NOTOT,7)
    IF(NOSTR)37,35,37
 37 NOSTR=NOSTR+1
    STIFF(KK,NOSTR)=XBRE
 35 IF(A(I,J)-9999.0)120,300,300
120 IF(NOSTR)121,122,121
122 NOSTR=1
    STIFF(KK,1)=X(I,J)
    STIFF(KK,2)=X(I,J)+A(I,J)/12.0*COS(ALPHA)
    IF(STIFF(KK,2)-BRAKE)123,300,300
123 NOSTR=2
    GO TO 300
121 IF(STIFF(KK,NOSTR)-X(I,J))125,300,124
124 IF(12.0*(STIFF(KK,NOSTR)-STIFF(KK,NOSTR-1))/COS(ALPHA)
   ¤-A(I,J)) 300,300,126
126 STIFF(KK,NOSTR)=STIFF(KK,NOSTR-1)+A(I,J)/12.0
   ¤*COS(ALPHA)
    IF(STIFF(KK,NOSTR)-X(I,J))127,300,300
127 STIFF(KK,NOSTR)=X(I,J)
    GO TO 300
125 STIFF(KK,NOSTR+1)=STIFF(KK,NOSTR)+A(I,J)/12.0
   ¤*COS(ALPHA)
    IF(STIFF(KK,NOSTR+1)-BRAKE)128,300,300
128 NOSTR=NOSTR+1
300 CONTINUE
    RETURN
    END
```

```
 SUBROUTINE STRESS (UBLO,UBLI,D,FLTO,FLTI,WIDI,WIDO,
□WEB,FBO,FBI,FA, FT,FY)
 CC=SQRT(2.0*3.1415926**2*29.5E06/FY)
 AREAO=WIDO*FLTO+(D/2.0-FLTO)*WEB
 RO=SQRT(FLTO*WIDO**3/12.0/AREAO)
 RULO=UBLO*12.0/RO
 CALL CSTRES(CC,RULO,FY,FAO)
 AREAI=WIDI*FLTI+(D/2.0-FLTI)*WEB
 RI=SQRT(FLTI*WIDI**3/12.0/AREAI)
 RULI=UBLI*12.0/RI
 CALL CSTRES(CC,RULI,FY,FAI)
 FA=(FAO*AREAO+FAI*AREAI)/(AREAO+AREAI)
 FT=0.6*FY
 AREAO=AREAO-D/3.0*WEB
 RO=SQRT(FLTO*WIDO**3/12.0/AREAO)
 RULO=UBLO*12.0/RO
 DLOA=UBLO*12.0*D/(FLTO*WIDO)
 H=D-FLTO-FLTI
 CALL BSTRES(CC,RULO,DLOA,FY,FLTO,H,WEB,WIDO,FBO)
 AREAI=AREAI-D/3.0*WEB
 RI=SQRT(FLTI*WIDI**3/12.0/AREAI)
 RULI=UBLI*12.0/RI
 DLOA=UBLI*12.0*D/(FLTI*WIDI)
 CALL BSTRES (CC,RULI,DLOA,FY,FLTI,H,WEB,WIDI,FBI)
 RETURN
 END
```

```
      SUBROUTINE UNIF (SPAN,H,SLOPE,BD,GD,X,Y,W,WC,PLGH,
     ¤NOCOL,NORAF, NOTOT,ERT,AXL,AXR,SHL,SHR,SML,SMR,R,ED)
      DIMENSION C(7)
      DIMENSION X(10,7),Y(10,7),WC(6),PLGH(10),ERT(10,7),
     ¤AXL(10,7), AXR(10,7),SHL(10,7),SHR(10,7),SML(10,7),
     ¤SMR(10,7),R(4)
      ALPHA=ATAN(SLOPE/12.0)
      COSA=COS(ALPHA)
      SINA=SIN(ALPHA)
      DIST=SPAN-(BD+2.0*GD)/12.0
      C(1)=1.0
      C(2)=4.0
      C(3)=2.0
      C(4)=4.0
      C(5)=2.0
      C(6)=4.0
      C(7)=1.0
      HP=SPAN/2.0*SLOPE/12.0
      R(3)=((WC(1)-WC(6))*H+HP*(WC(2)-WC(5)))*W
      R(2)=((H*(WC(1)-WC(6))*H/2.0+HP*(WC(2)-WC(5))*(H+HP
     ¤/2.0))/DIST +WC(3)*SPAN/8.0+3.0*WC(4)*SPAN/8.0)*W
      R(1)=(WC(3)+WC(4))*W*SPAN/2.0-R(2)
      DO 1 I=1,NOTOT
      DO 1 J=1,7
      SML(I,J)=R(1)*X(I,J)+R(3)*Y(I,J)-W*WC(3)*X(I,J)**2
     ¤/2.0
      SMR(I,J)=R(2)*X(I,J)-W*WC(4)*X(I,J)**2/2.0
      IF(I-NOCOL)2,2,3
    2 SML(I,J)=SML(I,J)-W*WC(1)*Y(I,J)**2/2.0
      SMR(I,J)=SMR(I,J)-W*WC(6)*Y(I,J)**2/2.0
      GO TO 1
    3 SML(I,J)=SML(I,J)-W*(WC(1)*H*(Y(I,J)-H/2.0)-WC(2)
     ¤*(Y(I,J)-H)**2 /2.0)
      SMR(I,J)=SMR(I,J)-W*(WC(6)*H*(Y(I,J)-H/2.0)-WC(5)
     ¤*(Y(I,J)-H)**2 /2.0)
    1 CONTINUE
      SUMM=0.0
      SUMY=0.0
      DO 5 I=1,NOTOT
      DO 5 J=1,7
      YDS=Y(I,J)*PLGH(I)/6.0/ERT(I,J)
      SUMM=SUMM+(SML(I,J)+SMR(I,J))*YDS*C(J)
    5 SUMY=SUMY+2.0*YDS*Y(I,J)*C(J)
      DS=ED*SLOPE/24.0
      SUMM=SUMM/3.0+(SML(NOCOL,7)+SMR(NOCOL,7))*(Y(NOCOL,7)
     ¤+DS/2.0) *DS/ERT(NOCOL,7)
      SUMY=SUMY/3.0+2.0*(Y(NOCOL,7)+DS/2.0)**2*DS/ERT(NOCOL,
     ¤7)
      SUMM=SUMM+(SML(NOCOL+1,1)+SMR(NOCOL+1,1))*(Y(NOCOL+1,
     ¤1)- DS/2.0*SINA)*DS/ERT(NOCOL+1,1)
      SUMY=SUMY+2.0*(Y(NOCOL+1,1)-DS/2.0*SINA)**2*DS
```

```
 □/ERT(NOCOL+1,1)
  R(4)=-SUMM/SUMY
  R(3)=R(3)+R(4)
  DO 6 I=1,NOTOT
  DO 6 J=1,7
  SML(I,J)=SML(I,J)+Y(I,J)*R(4)
6 SMR(I,J)=SMR(I,J)+Y(I,J)*R(4)
  DO 7 I=1,NOCOL
  DO 7 J=1,7
  AXL(I,J)=-R(1)
  AXR(I,J)=-R(2)
  SHL(I,J)=R(3)-W*WC(1)*Y(I,J)
7 SHR(I,J)=R(4)-W*WC(6)*Y(I,J)
  DO 8 M=1,NORAF
  I=NOCOL+M
  DO 8 J=1,7
  HORL=R(3)-W*(WC(1)*H+WC(2)*(Y(I,J)-H))
  HORR=R(4)-W*(WC(6)*H+WC(5)*(Y(I,J)-H))
  VRTL=-R(1)+W*WC(3)*X(I,J)
  VRTR=-R(2)+W*WC(4)*X(I,J)
  AXL(I,J)=HORL*COSA+VRTL*SINA
  AXR(I,J)=HORR*COSA+VRTR*SINA
  SHR(I,J)=HORR*SINA-VRTR*COSA
8 SHL(I,J)=HORL*SINA-VRTL*COSA
  RETURN
  END
```

```
      SUBROUTINE UNSUL(X,Y,NOGRT,GIRT,NOPUR,PURL,NOCOL,
     ¤NOTOT,UBLI,UBLO, H,GD,SLOPE,D,BD,SPAN,FIRST)
      DIMENSION X(10,7),Y(10,7),GIRT(10),PURL(20),D(10,7),
     ¤UBLI(10,7), UBLO(10,7)
      ALPHA=ATAN(SLOPE/12.0)
      SINA=SIN(ALPHA)
      COSA=COS(ALPHA)
      BETA=1.5707963-ALPHA/2.0
      TANB=SIN(BETA)/COS(BETA)
      IF(Y(NOCOL,7)-GIRT(NOGRT))1,1,2
    2 IF(NOGRT-10)3,4,4
    4 WRITE(3,5)
    5 FORMAT(' EXCESSIVE NUMBER OF GIRTS')
      STOP 115
    3 NOGRT=NOGRT+1
      GIRT(NOGRT)=H-GD/12.0*TANB
    1 DO 6 I=1,NOCOL
      DO 6 J=1,7
      IF(I+J-2)8,7,8
    7 UBLO(1,1)=0.0
      UBLI(1,1)=0.0
      GO TO 6
    8 IF(Y(I,J)-GIRT(1))9,10,10
    9 UBLO(I,J)=GIRT(1)
      UBLI(I,J)=GIRT(1)
      GO TO 6
   10 DO 11 K=2,NOGRT
      IF(Y(I,J)-GIRT(K))12,13,11
   11 CONTINUE
      STOP 11
   13 UBLO(I,J)=0.0
      UBLI(I,J)=0.0
      GO TO 6
   12 UBLO(I,J)=GIRT(K)-GIRT(K-1)
      IF(Y(NOCOL,7)-GIRT(K))14,16,16
   14 UBLI(I,J)=Y(NOCOL,7)-GIRT(K-1)
      GO TO 6
   16 UBLI(I,J)=UBLO(I,J)
    6 CONTINUE
      UBLI(NOCOL,7)=0.0
      REF=SPAN/2.0-BD/24.0-GD/12.0
      M=NOCOL+1
   17 DO 18 I=M,NOTOT
      DO 18 J=1,7
      DO 19 K=1,NOPUR
      IF(X(I,J)-D(I,J)/24.0*SINA-(REF-PURL(K)))19,21,22
   19 CONTINUE
      STOP 125
   21 UBLO(I,J)=0.0
      UBLI(I,J)=0.0
      GO TO 18
```

```
22 IF(K-1)23,24,23
24 UBLO(I,J)=FIRST/6.0
   UBLI(I,J)=FIRST/6.0
   GO TO 18
23 UBLO(I,J)=(PURL(K)-PURL(K-1))/COSA
   IF((REF-PURL(K))-(X(M,1)-D(M,1)/24.0*SINA))25,26,26
25 UBLI(I,J)=(REF-PURL(K-1)-(X(M,1)-D(M,1)/24.0*SINA))
  ¤/COSA
   GO TO 18
26 UBLI(I,J)=UBLO(I,J)
18 CONTINUE
   UBLI(M,1)=0.0
   RETURN
   END
```

```
      SUBROUTINE WIND(SLOPE,H,SPAN,C)
      DIMENSION C(6)
      HOW=H/SPAN
      IF(HOW-1.0)1,1,2
    2 HOW=1.0
    1 IF(HOW-0.525)3,3,4
    3 C(1)=0.7
      GO TO 5
    4 C(1)=0.7+(HOW-0.525)/4.75
    5 IF(HOW-0.25)6,6,7
    6 C(2)=(-0.2292+0.0592*SLOPE)
      GO TO 8
    7 C(2)=(-0.06503-0.6979*HOW+0.05327*SLOPE+0.022286
     ¤*SLOPE*HOW)
    8 IF(C(2))9,10,10
   10 IF(C(2)-0.6)11,11,12
   12 C(2)=0.6
      GO TO 11
    9 IF(HOW-0.25)13,13,14
   13 C(2)=(-4.41077*HOW+1.0234*HOW*SLOPE)
      GO TO 15
   14 IF(HOW-0.6)17,18,18
   17 C(2)=(-1.06642+0.3789*SLOPE-0.6507*SLOPE*HOW+0.4027
     ¤*SLOPE*HOW**2)
      GO TO 15
   18 C(2)=(-1.07039+0.2157*ALPHA-0.1411*ALPHA*HOW+0.0384
     ¤*ALPHA*HOW**2)
   15 IF(C(2)+0.6)16,11,11
   16 C(2)=(-0.6)
   11 C(3)=C(2)
      C(4)=(-0.5)
      C(5)=(-0.5)
      C(6)=(-0.4)
      RETURN
      END
```

BIBLIOGRAPHY

LITERATURE CITED

(1) L. S. Beedle, A. D. Freas, P. H. Karr, J. D. Haltiwanger, and Joseph Penzien, "Research Needs in Structural Engineering for the Decade 1966-1975," Report of the Structural Division Research Committee, American Society of Civil Engineers, January 1966, pp. 3-9.

(2) J. M. Bourgeois, A Problem Oriented Language for the Optimum Plastic Design of a Gabled Frame, Unpublished M. S. Thesis, Mississippi State University, 1968, 84 pp.

(3) Steven J. Fenves, "STRESS - A Computer Programming System for Structural Engineering Problems," Technical Report T63-2, Department of Civil Engineering, Massachusetts Institute of Technology, June 1963, 48 pp.

(4) G. G. Goble, "Structural Synthesis in Cost-Oriented Structural Design," Building Research, Building Research Institute, March - April 1966.

(5) J. Keifer, "Sequentia Minimax Search for a Maximum," Proceedings American Mathematical Society, vol. 4, 1953, pp. 503-506.

(6) A. J. McNair, "Electronic Computers in Surveying," Transactions American Society of Civil Engineers, vol. 124, 1959, p. 252.

(7) C. L. Miller, "ICES COGO," Research Report T65-9, Department of Civil Engineering, Massachusetts Institute of Technology, August 1965.

(8) C. L. Miller and D. Ross, "Man Machine Communications in Civil Engineering," Preliminary Notes for the Conference on Civil Engineering Computer Systems, Massachusetts Institute of Technology, July 1966, p. 32.

(9) C. L. Miller and R. A. Walter, "Communicating with Computers in Civil Engineering Design," Technical Report T65-4, Department of Civil Engineering, Massachusetts Institute of Technology, March 1965, 23 pp.

(10) R. Razani and G. G. Goble, "Optimum Design of Constant Depth Girders," Journal of the Structural Division, Proceedings American Society of Civil Engineers, vol. 92, no. ST2, April 1966, pp. 253-281.

(11) D. Roos, <u>ICES System Design</u>, The MIT Press, 1966, 239 pp.

(12) Douglas T. Ross, "The AED Approach to Generalized Computer-Aided Design," <u>Proceedings of Twenth-second National Conference, Association of Computing Machinery Publication</u>, 1967, pp. 367-385.

(13) M. F. Rubinstein and J. Karegozian, "Building Designs Using Linear Programming," <u>Proceedings of the American Society of Civil Engineers</u>, vol. 92, ST6, December 1966, pp. 223-245.

(14) Douglas J. Wilde, <u>Optimum Seeking Methods</u>, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1964, 193 pp.

VITA

Marion Willson Corey was born in Jackson, Mississippi, February 11, 1932, the son of Lyle Vernon and Nema Gaddis Corey. After being graduated from Meridian High School, Meridian, Mississippi, he attended Meridian Junior College for one year. He entered Alabama Polytechnic Institute and received his Bachelor of Science in 1954. After serving in the United States Navy from 1954 until 1957, he held the position of Instructor of Civil Engineering at Mississippi State University and received the degree of Master of Science in Civil Engineering in 1960. During 1960 and 1961 he held the rank of Assistant Professor at Mississippi State. In the summer of 1961 he attended the National Science Foundation Summer Institute at Louisiana Polytechnic Institute. In 1961 he entered the Georgia Institute of Technology to begin work toward the degree Doctor of Philosophy. He was an instructor at Southern Technical Institute while in residence at Georgia Tech. Mr. Corey has been a consultant to several engineering firms from 1959 to the present time. In September, 1963, he returned to Mississippi State University as Associate Professor of Civil Engineering. He presently holds the rank of Lieutenant Commander in the United States Naval Reserve and is the Commanding Officer of Naval Reserve Surface Division 6-21, Starkville, Mississippi.